

Collaborative Environments for Disaster Relief

by

Yusuke Mizuno

M.Eng., Civil Engineering
University of Tokyo, 2000

Submitted to the Department of Civil and Environmental Engineering
In Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Civil and Environmental Engineering

at the

Massachusetts Institute of Technology

June 2001

© 2001 Yusuke Mizuno, All rights reserved

The author hereby grants to MIT permission to reproduce
and to distribute publicly paper and electronic
copies of this thesis document in whole or in part.

Signature of Author:

Department of Civil and Environmental Engineering

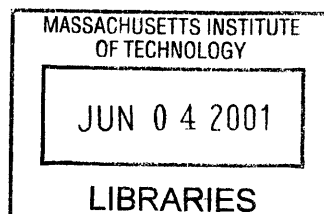
May 11, 2001

Certified by:

Feniosky Peña-Mora
Associate Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by:

Oral Buyukozturk
Chairman, Departmental Committee on Graduate Studies



ENG

Collaborative Environments for Disaster Relief

by

Yusuke Mizuno

Submitted to the Department of Civil and Environmental Engineering
on May 11, 2001 in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Civil and Environmental Engineering

ABSTRACT

In a large-scale disaster relief mission, a variety of personnel and organizations with different expertise take part in the relief activities towards a common goal. Interactions between them are important to promote efficiency, and the relief activities, especially search and rescue (SAR) activities, are carried out in a short time period.

Some Emergency Offices of City/State/Federal governments and two case studies of actual disaster relief activities after large earthquakes were examined in order to recognize the issues in terms of collaborative environments during a disaster relief missions. These examinations have revealed the fact that the availabilities of site-specific information is a major concern when some decisions concerning relief activities are made, but the communication between on-site personnel and headquarters is not enough mainly due to the lack of cross-organizational coordination.

Several commercial solutions have been developed and several studies have been made on collaborative environments at MIT. The current typical solutions consist of e-mail system, group discussion, bulletin board system, and shared data repositories integrated with mapping systems. Some further enhancements of these systems should be considered: multiple client access and detail design of thin portable client devices.

To develop collaborative environments for disaster relief missions, the requirements have been gathered and the requirement analysis has been developed. According to functional role, relief mission participants are classified into four groups: field agent, team leader, coordinator, decision maker and specialist/advisor. The category “field agent” plays an integral role during a mission so that its activities are focused on. The hierarchy of collaborative session has been developed on account of that of participant roles.

The multiple device access is an important functionality of collaborative environments. Some of the leading portable devices are featured and three types of wireless communication are described. To collect and report site-specific information in a disaster area is critical to make decisions. The needs and requirements of robot-human interaction that can be applied to a relief mission are discussed in comparison with SAR dogs.

Thesis supervisor: Feniosky Peña-Mora

Title: Associate Professor of Civil and Environmental Engineering

ACKNOWLEDGEMENT

I would like to thank Professor Feniosky Peña-Mora for his detailed comments, suggestions, and constant support during the entire time duration of this study. I am also grateful to Chang Kuang and Wei Wang for their advice.

TABLE OF CONTENTS

1. Introduction.....	10
1.1 Background	10
1.2 Problems	11
1.3 Objectives	12
1.4 Approach	13
2. Disaster Relief Mission	14
2.1 Search and Rescue Mission	14
2.2 Local Operations: San Francisco.....	15
2.3 Local Operations: New York City	16
2.4 Federal Operations: FEMA	16
2.5 Large-Scale Disaster: Kobe Earthquake.....	18
2.5.1 Damage Description	18
2.5.2 SAR Activities	19
2.6 Large-Scale Disaster: Turkey Earthquake.....	21
2.6.1 Damage Description	21
2.6.2 SAR Activities	22
3. Collaborative Environment for Emergency Management.....	23
3.1 EIS/GEM Infobook.....	23
3.2 Response Information Management System (RIMS).....	26
3.3 E Team.....	28
3.4 Lotus Solutions	31
3.4.1 Lotus Domino	33
3.4.2 Lotus Notes.....	34
3.4.2 Mobile Access to Lotus Services.....	35
3.5 Collaborative Environment at MIT.....	37
3.5.1 Meeting structure.....	37
3.5.2 Information Policies.....	38
3.5.3 Meeting Protocols	38
3.6 Collaborative Environment at Draper	39
3.6.1 Mission Planning Applications.....	40
3.6.2 Draper Application Architecture and Data Persistence	40

3.6.3 Multiple Device Interface.....	41
3.7 Collaboration between MIT and Draper	42
4. Emergency Management System Development	43
4.1 Scenario.....	43
4.1.1 Participants	43
4.1.2 Role Description.....	46
4.2 System Overview	48
4.2.1 Device and Network Limitation	48
4.2.2 System Overview Description	49
4.3 Use Case.....	50
4.3.1 Glossary	50
4.3.2 Diagram	51
4.4 Collaborative Environment Data Flow.....	54
4.5 Collaboration System Requirements.....	57
4.5.1 Structured Collaborative Environment.....	57
4.5.2 Representation of Collaborators	61
4.5.3 Abstracted Information Policy and Protocol Enforcement	62
4.5.4 Persistence of Collaboration Data	62
4.6 System Architecture	63
4.6.1 Client Application	66
4.6.2 Information Policy	70
4.6.3 Data Repository	72
4.6.4 Collaboration Manager	73
4.6.7 Custom Components.....	75
5. Multiple Device Access.....	77
5.1 Desktop Environment.....	77
5.2 Portable PC	77
5.3 Personal Digital Assistant.....	80
5.4 Java-Enable Phone	82
5.4.1 Java 2 Platform, Micro Edition	82
5.4.2 Java-enabled Phone Service	83
5.4 Wireless Communication	84

5.4.1 Wireless WAN	85
5.4.2 Wireless LAN.....	86
5.4.3 Wireless PAN.....	86
6. Robot-Human Interaction.....	87
6.1 Overview.....	87
6.2 Requirements.....	87
6.3 LEGO MINDSTORMS	88
6.3.1 Programming Environment.....	89
6.3.2 Input: Sensors	89
6.3.3 Output.....	89
6.4 Jini Technology	89
7. Conclusion	91

LIST OF FIGURES

Figure 1.1: Number of people rescued/searched by SAR teams (Kobe Earthquake)	11
Figure 3.1: EIS/GEM Infobook [10]	25
Figure 3.2: EIS/GEM resource inventory [10]	26
Figure 3.3: California's Response Information Management System [11]	27
Figure 3.4: E Team Internet-based Emergency Management Software [13]	30
Figure 3.5: Information cross-referenced to a map (E Team) [13]	30
Figure 3.6: Lotus Notes and Domino components [14]	33
Figure 3.7: CAIRO Meeting Environment [16]	39
Figure 3.8: Geospatial Application [17]	41
Figure 3.9: Temporal Synchronization Application [17]	42
Figure 4.1: Disaster relief mission scenario.....	46
Figure 4.2: Collaborator use case diagram.....	51
Figure 4.3: Field agent use case diagram	52
Figure 4.4: Field agent use case diagram (collect info.)	53
Figure 4.5: Session participant use case diagram.....	53
Figure 4.6: Collaboration session chairperson use case diagram	54
Figure 4.7: Specialist/advisor use case diagram.....	54
Figure 4.8: Collaborative environment data flow.....	55
Figure 4.9: Hierarchy of collaboration sessions	56
Figure 4.10: Class diagram.....	57
Figure 4.11: Sample session organization structure	59
Figure 4.12: Protocol handling of speech request.....	61
Figure 4.13: Placement of information policy	62
Figure 4.14: Model-view-controller architecture.....	65
Figure 4.15: Four components of the collaboration system architecture.....	66
Figure 4.16: Sample client application configuration.....	68
Figure 4.17: Transaction paths for client application	69
Figure 4.18: Database structure for collaboration session information.....	71
Figure 4.19: Information policy implementation	71
Figure 4.20 Repository manager and data repository	73
Figure 4.21: Collaboration manager architecture	75

Figure 4.22: Custom implementation of system components	76
Figure 5.1: J2ME wireless architecture [24]	83

LIST OF TABLES

Table 2.1: Earthquake victims and damage to buildings (Kobe Earthquake)..... 19

Table 2.2: Rescued people by Fire service and JGSDF (Kobe Earthquake)21

Table 4.1: Participants of disaster relief mission 45

Table 4.2: Client device 45

Table 5.1: Portable PC 79

Table 5.2: PDA..... 81

Table 5.3: Java-enable phone 84

1. INTRODUCTION

Due to the nature of disaster relief missions, distributed teams work together and the interaction between the teams is critical to achieve their projects and missions. Site-specific information is essential to make a decision on each phase. This thesis focuses on collaboration among disaster relief mission participants, investigates the requirements and designs the architecture of an Emergency Management System for unstable and hostile environments.

1.1 BACKGROUND

A disaster relief mission relies on geographically distributed teams that consist of several roles of personnel such as field agents, team leaders, coordinators, decision makers and specialists/advisors. The teams are comprised of various individuals and organizations with different expertise usually from foreign countries. Several constraints exist in this mission: one of the biggest issue is that the mission needs to be addressed in a short amount of time; cultural, discipline differences may exist since a disaster relief mission is performed by participants from various organization or countries; on-site information is often vague and/or unpredictable. The collaboration among the teams requires a platform that facilitates communication and decision-makings toward a common goal.

One aspect of a large-scale disaster relief mission is that activities such as searching and rescuing survivals, and repairing buildings temporarily to support rescue missions must be carried out in a short time period. Searching and rescuing survivals should be performed immediately after a disaster occurs. Figure 1.1 shows a relation between time and the number of people rescued after the Great Hanshin-Awaji (Kobe) Earthquake[4][5][6] that occurred on January 17th, 1995. The number of survivals decreased rapidly after the day the earthquake hit.

The other aspect is that various kinds of teams and organizations take part in search and rescue activities. When Kobe Earthquake occurred, fire fighting rescue teams from other cities as well as from local cities, the police and the Self-Defense Forces were in charge of rescuing survivals. The Switzerland government sent rescue teams which

consisted of 25 search and rescue (SAR) team members and 3 SAR dogs. The France government provided 60 SAR members and 4 SAR dogs. An NGO in the U.K. provided 15 SAR members and several kinds of SAR equipment. When a large earthquake occurred in Turkey on August 17th, 1999, forty-six countries provided assistance to Turkey according to the United Nations Development Programme (UNDP). Some of this assistance included 2,276 SAR team members, 224 SAR dogs, 23 ambulances, and 709 medical personnel. The U.S. Agency for International Development (USAID)/Office of U.S. Foreign Disaster Assistance (OFDA) deployed two SAR teams to help recover earthquake survivors.

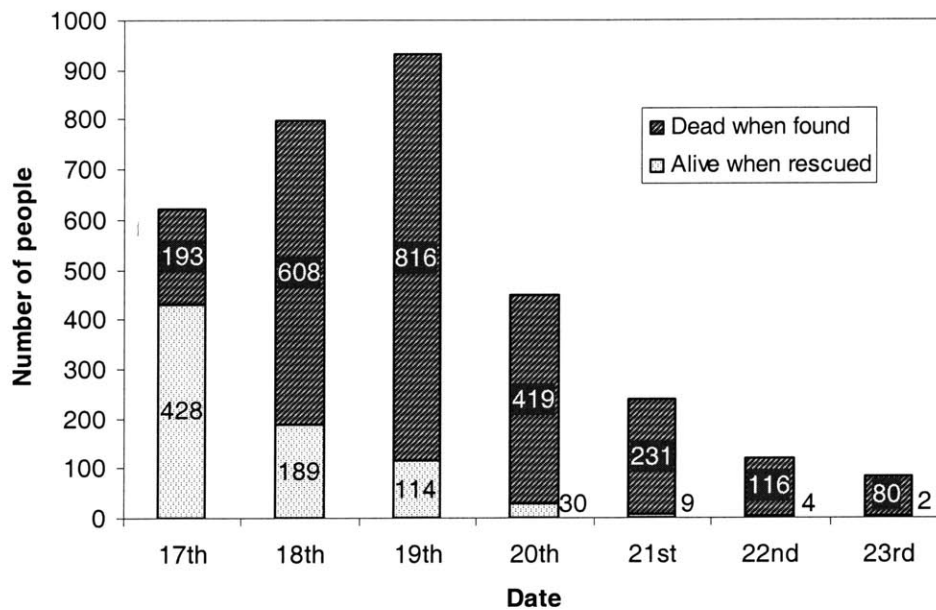


Figure 1.1: Number of people rescued/ searched by SAR teams (Kobe Earthquake)

1.2 PROBLEMS

Two major problems arise: how to organize collaboration among participants; how to maintain information related to the disaster. Each participant has different issues and deals with different site-specific environments. Several participants need to collaborate with each other in terms of the same issue such as a particular building collapse, and form a group of personnel. When a large-scale disaster occurs, a large number of groups of participants with

the particular topic exist simultaneously. Such groups may require the other participants or may synchronize with other groups. The collaborative platform should be designed to meet with the requirement addressed in a disaster relief mission.

To promote an effective disaster relief mission, information related to a disaster should be accurate and reliable, which means that information needs to be maintained in a consistent way so that every participant can access to the latest information. The system must keep track of information transaction in order to make the information be secure.

In a disaster area, two kinds of information are critical for decision-makings, resource allocation and site-specific information. The resource allocation consists of staff deployment and equipment allocation. The resource allocation changes frequently since it is difficult to estimate the need of the resources beforehand and actual site-specific information triggers the new deployment. Site-specific information is usually vague and becomes distinct gradually as the on-site activities progress.

Some of the devices and technologies have potential for facilitating a disaster relief mission. Since most of the critical information is related to field agents, a client device and tool for a field agent should be investigated and the interface between these devices and the whole system should be designed. This study proposes to introduce wireless devices and an autonomous robot. Wireless devices are suitable to establish connection between on-site personnel and the collaboration platform and an autonomous robot is adopted to collect information about disaster environment where personnel cannot enter.

1.3 OBJECTIVES

This study designs an emergency management system that consists of collaboration platform and data repositories. The system facilitates the collaboration between the participants and allows multiple device accesses so that they can share the information and make decisions efficiently. The collaboration platform furnishes a virtual conference room, and offers the privilege to organize a meeting and the methods of sharing the information. The data repositories keep tracks of resources related to a disaster relief mission so that these

resources can be re-allocated efficiently, and store site-specific information to be accessed by every participants.

A method of gathering site-specific information is developed by focusing on a robot that can drive itself around according to the instruction and can sense the nearby environment including the communication between a robots and a PC/handheld device.

1.4 APPROACH

In Chapter 2, general description about a disaster relief mission is provided. It involves several emergency management systems of government organizations and actual large-scale disaster relief missions.

Chapter 3 describes several existing solution of emergency management. The first part features the commercial software that governmental organizations adopted as their emergency management systems. The latter part explains about collaboration environment developed at MIT.

Chapter 4 introduces the feature of the collaborative system for disaster relief missions. First, target users and activities are defined by developing use cases of the system and then the architecture is designed using UML diagrams.

Chapter 5 focuses on client devices that allow users to access web-based resources and to participate in collaborative environments in different ways. Much attention is paid to portable/ handheld devices such as PDAs and Java-enable phones. Also this chapter features wireless networks.

Chapter 6 introduces human-robot interaction in order to collect site-specific information primary addressing the issue of locating victims. The potential needs for robots and requirements of human-robot interaction are discussed.

2. DISASTER RELIEF MISSION

2.1 SEARCH AND RESCUE MISSION

This section describes an overview of a search and rescue (SAR) activities in the U.S. When a large-scale disaster occurs:

- Response always begins at the local level. Local fire departments, emergency management, and local and state law enforcement are the first arrive at the disaster area and begin SAR and response.
- Following the disaster the local emergency manager may request assistance from the state; if the need is great, the state may in turn requests federal assistance; in the event of a large-scale disaster, FEMA deploys three of the closest task forces.
- After arriving at the site, structural specialist, who are licensed professional engineers charged with making the rescue safe for the rescuers. They provide direct input to the SAR team members about structural integrity of the building and the risk of secondary collapses.
- Heavy equipment is used to remove large rubble and debris that could injure rescue workers and impede rescue operations.
- The search team ventures around and into the collapsed structure, carefully shoring up structures while attempting to locate trapped victims. Rescuers use electronic listening devices, extremely small search cameras and specially trained search dogs to help locate victims.
- Once a victim is located, the search group begins the daunting task of breaking and cutting through thousands of pounds of concrete, metal, and wood to reach the victims. They also stabilize and support the entry and work areas with wood shoring

to prevent further collapse.

- Throughout the effort hazardous materials specialists evaluate the disaster site, and decontaminate rescue and medical members who may be exposed to hazardous chemicals or decaying bodies.
- Heavy rigging specialists direct the use of heavy machinery, such as cranes and bulldozers. These specialists understand the special dangers of working in a collapsed structure, and help to ensure the safety of the victims and rescuers inside.
- Technical information and communication specialists ensure that all team members can communicate with each other and the task force leaders, facilitating search efforts and coordinating evacuation in the event of a secondary collapse.
- Logistics specialists handle the more than 16,000 pieces of equipment to support the search and extrication of the victims. The equipment cache includes such essentials as concrete cutting saws, search cameras, medical supplies, and tents, cots, food and water to keep the task force self-sufficient for up to four days.

2.2 LOCAL OPERATIONS: SAN FRANCISCO

The Mayor's Office of Emergency Service (MEOS)[2] is the component of the Mayor's staff with responsibility for strategic emergency planning. MEOS develops and manages the emergency plan for the City and County of San Francisco, coordinates all protective and relief service, trains emergency service.

The Emergency Operation Center (EOC) is the central location for coordination of emergency response activities. EOC serves as the location where damage information is assessed, the over all disaster situation is analyzed, and the City's response and recovery priorities are established.

The San Francisco Office of Emergency Services - Auxiliary Communications Service

(ACS) serves as an alternate means of communications for the Office of Emergency Services.

2.3 LOCAL OPERATIONS: NEW YORK CITY

The Mayor's Office of Emergency Management (OEM)[3] is the On-Scene Coordinator at all emergency incidents and has the responsibility for interagency coordination and communication. When a disaster occurs, OEM activates the City's Emergency Operations Center, and administers the Urban Search and Rescue Team.

During an emergency, the Watch Command officers monitor all emergency services radio frequencies (Police, Fire, EMS), and have access to NYSPIN (New York State Police Information Network); NAWAS (The National Warning and Alert System, which offers information provided by the Federal Emergency Management Agency and the New York State Emergency Management Office); The National Weather Service; The Emergency Medical Service's Computer-Aided Dispatch, and the Con Edison Hotline. Monitoring events and developing situations city-wide, these officers keep the OEM staff informed, act as a response dispatch and communications link, and provide inter-agency notification services.

If an emergency requires a multi-agency response, OEM is the on-scene coordinator, working in conjunction with the Incident Commander (the highest ranking chief officer of either the police department or fire department). OEM will also set up a unified interagency command post, and provide support to the operation by streamlining the procurement of resources.

The Emergency Operation Center offers space and facilities where the related agencies can operate their tasks: they can interact with their home agency command centers; they can collaborate with the administrative staff of OEM.

2.4 FEDERAL OPERATIONS: FEMA

The Federal Emergency Management Agency (FEMA)[1] is an independent agency of the federal government, reporting to the President. The goal of FEMA is to reduce loss of

life and property and protect the nation's critical infrastructure from all types of hazards.

The National Urban Search and Rescue (US&R) Response System, established under the authority of FEMA, is a framework for the structuring local emergency services personnel into integrate disaster response task forces. These task forces, complete with necessary tools and equipment, and required skills and techniques, can be deployed by FEMA for the rescue of victims of structural collapse.

Disaster may require resources beyond the capabilities of the local or State authorities. In response of regional requests for support, FEMA provides: mobile emergency response support; task forces comprised of SAR operation staff, equipment, and SAR dogs; incident support teams that supports task forces mission through logistical, electronic and coordination expertise.

In a disaster area, FEMA provides mobile telecommunications, logistics, and operational support to the Federal, State, and Local responders. The Response and Recovery Directorate's Mobile Operations Division (RR-MO) maintains mobile telecommunications, operational support, life support, and power generation assets for the on-site management of disaster relief activities.

There are 28 FEMA US&R Task Forces spread throughout the continental United States trained and equipped by FEMA to conduct round-the-clock SAR operations following disasters such as earthquakes, tornados, floods, hurricanes, aircraft accidents, hazardous materials spills and catastrophic structure collapses. Each task force consists of two 31-person teams, four canines, and a comprehensive equipment cache. Task Force members work in four areas of specialization: search, to find victims trapped after a disaster; rescue, which includes safely digging victims out of tons of collapsed concrete and metals; technical, made up of structural specialists who make rescues for the rescuers; and medical, which cares for the victims before and after a rescue. Emergency managers with special training in urban SAR environments conduct SAR activities.

The task force equipment cache must be a construction site, communication center, high-tech engineering firm, mobile emergency room, and camp rolled into one. The cache

consists of five types of equipment: medical, rescue, communications, technical support and logistics. The communication section allows rescuers to stay in contact in case of a find or an evacuation. Generators, lights, radios, cellular phones, laptop computers and other electronics equipment are used.

2.5 LARGE-SCALE DISASTER: KOBE EARTHQUAKE

The Great Hanshin-Awaji Earthquake[4][5][6] (also called Kobe Earthquake) of magnitude (M) 7.2 on January 17, 1995, jolted central western Japan and brought the most serious damage in the last 70 years. The earthquake hit the southern part of Hyogo Prefecture including the city of Kobe, Japan's sixth largest city with 1.5 million population, and the northern part of Osaka Prefecture. Death toll from the earthquake rose over five thousand, and about 100,000 houses and buildings were totally destroyed. It was really a big shock to various societies in Japan.

2.5.1 Damage Description

The earthquake brought serious damage of 6,432 dead people, 3 missing and 4,3792 injured. The victims and damage to buildings are summarized in Table 2.1. Not only strong shaking but also wide spread fires after the earthquake were the main cause of building damage and human casualties. Civil infrastructures were also severely affected: a large number of highway and railway bridges were collapsed; minor to severe damage occurred in 5 thermal power stations, 17 substations, 23 transmission lines, and several hundred distribution lines; gas supply was stopped for 860 thousand customers of Osaka Gas; raw water pipes from the river to filtration plants were broken in many places and some filtration plants were also damaged; structural damage to telecommunication systems was large, mostly on lines and buried ducts.

Table 2.1: Earthquake victims and damage to buildings (Kobe Earthquake)

Victims	Dead		6,432	
	Missing		3	
	Injured	Heavily injured	8,782	
		Lightly injured	35,010	
Buildings	House/Residential Building		Fully Collapsed	104,906
			Half Collapsed	144,272
			Partially Collapsed	263,702
	Non-residential Building	Public Facilities	Collapsed	865
		Other	Collapsed	3983

2.5.2 SAR Activities

After the earthquake occurred at 5:46 in the morning of January 17, 1995, immediate responses in the heavily hit areas were concentrated to search and rescue, and fire fighting. Hyogo Police Department established disaster guard headquarters at 6:15 a.m., Hyogo Prefectural government and Kobe city independently established their emergency response offices at 7:00 a.m. National government decided to establish major emergency response headquarters during the cabinet meeting in the morning, and the first meeting of the headquarters held in the afternoon agreed to help municipal governments in the heavily affected area with respect to rescue operations and earliest possible recovery.

But information was scarce and resources were lacking in the affected area. The first personnel reported to the prefectural office at 6:45, an hour after the earthquake, and at 2:00 p.m., more than eight hours after the earthquake, only about 20 percent of the prefectural personnel could report to their head office. The request for help to the Japan Ground Self Defense Force (JGSDF) was made by the prefectural office at 10:00.

The Governor of Hyogo prefecture requested the Fire Defense Agency, a national

organization in Tokyo, for help. It is reported that during the day of the earthquake 1,180 personnel from 19 local governments reached Kobe with 243 vehicles. However, lack of water as well as extremely heavy traffic congestion hindered efficient firefighting operations. By the evening of the day, most small-scale fires were extinguished, but eleven big fires were still raging. These fires were mostly under control on the morning of January 18 after having burnt down about 54 hectares. The total burnt-down area is reported as about 66 hectares.

Search and rescue (SAR) was done by the people from the fire department, JGSDF, and the people from the police department. However, it seems that by far the majority of the people trapped under collapsed houses were rescued by family members and/or neighbors. An estimate tells that a total of about 20,000 people were buried under collapsed houses, of which about 15,000 were rescued by the ordinary people.

The JGSDF headquarters located in the peripheral area of the stricken region was ready to mobilize its troops at 6:00 a.m. They waited for the official request of the prefectural governor to come, which is generally required by law. It was 10:00 a.m., about four hours after the earthquake, that the governor of Hyogo prefecture officially requested help to JGSDF. However, because of the lack of information on the scale of assistance, the final decision to send a large troop to the stricken area was made at 3:00 a.m. on the early morning of January 18, and it was 6:00 p.m. of the same day, about one and half day after the occurrence of the earthquake, that a large troop of 9,000 reached the heavily stricken area.

It is believed that the mobilization of JGSDF was too late. Looking at the numbers of the people rescued by fire service people and JGSDF on January 17 and 18 (Table 2.2), it is inferred that, if JGSDF had been mobilized earlier, more people could have been rescued alive.

There are more than several reasons for the delay of the dispatch of JGSDF. The Self Defense Force Law is said to be inadequate in that a request of the prefectural governor is required before JGSDF troops be dispatched. It is also said that the relationship between the local government and the JGSDF in the area should have been better. Some also say that JGSDF itself had not been prepared for such a large-scale disaster and that they did

not have efficient rescue equipment during the immediate hours. However, the greatest problem seems to be the lack of information regarding the real status of the disaster. If it had been known that a disaster with 5,500 deaths was taking place, the responses of all organizations must have been different.

Table 2.2: Rescued people by Fire service and JGSDF (Kobe Earthquake)

		17th Jan.	18th	19th	20th	21st	22nd	23rd	24th - 3rd Feb.	Total
Alive	FIRE SERVICE	396	123	70	18	6	4	2	0	619
	JGSDF	32	66	44	12	3	0	0	0	157
	Total	428	189	114	30	9	4	2	0	776
Dead	FIRE SERVICE	137	320	385	185	146	86	14	20	1293
	JGSDF	56	288	431	234	85	30	66	27	1217
	Total	193	608	816	419	231	116	80	47	2510
Dead and alive		621	797	930	449	240	120	82	47	3286
Survival rate		69%	24%	12%	6.7%	3.8%	3.3%	2.4%	0%	23.6%

2.6 LARGE-SCALE DISASTER: TURKEY EARTHQUAKE

An earthquake of M=7.8 hit Turkey on August 17, 1999 under the city of Izmit. An aftershock of M=5.8 followed on September 13. The second major earthquake of M=7.2 hit northwestern Turkey near the town of Duzce (located on the northern coast of Turkey, east of Istanbul) on November 12. The earthquakes wreaked havoc on the region, leaving thousands dead, and hundreds of thousands injured or homeless.

2.6.1 Damage Description

As of September 6, 1999 casualties total 15,135 confirmed dead and 23,984 injured,

with additional thousands missing and presumed dead. Turkey's National Security Council estimated 200,000 people were made homeless.

The earthquake damaged buildings across seven provinces for a distance of 250 km from Istanbul to Bolu. As many as 70% of the buildings in portions of the cities of Adapazari, Golcuk, Izmit, Topcular, and Kular were severely damaged or collapsed (Coburn, Halling and Sezen, 1999). Nearly all the fatalities and injuries can be attributed to building collapse. As of September 6, 1999, the Government Crisis Center reported 20,957 buildings were heavily damaged or collapsed. Other reports suggest that up to 115,000 buildings were damaged beyond repair. Building losses are reported to amount to about US\$5 billion.

After the November 12 earthquake, most of the damage was concentrated in Kaynasli a small town on the main highway between Duzce and Bolu. Loss of life in Duzce seemed to concentrate in few collapsed buildings that were "lightly" damaged in the August 17 earthquake, superficially repaired and later inhabited. Loss of life was somewhat limited since the buildings damaged in the August 17 earthquake were already vacated. Total number of heavily damaged buildings was in the vicinity of 200. 550 people lost their lives (as of November 17, 1999) and over 3000 people were hospitalized with injuries.

2.6.2 SAR Activities

After the August 17 earthquake, initial search and rescue (SAR) was strongly criticized as slow and unorganized. This appeared to be the result of loss of communications, lack of command and control, shortage of equipment and materials, and an absence of disaster response training. Much of the initial response was by survivors. International SAR teams began arriving within 24 hours, and 65 foreign SAR teams saved 621 lives in intensive efforts that continued for four or five days following the earthquake. Initially there was friction between SAR teams and heavy equipment operators attempting to aid in the search. SAR workers were concerned that additional injuries would be caused by heavy equipment, but later realized the enormity of the task and accepted the method. Not until three days after the event did 50,000 soldiers arrive to assist in SAR and debris clearance.

3. COLLABORATIVE ENVIRONMENT FOR EMERGENCY MANAGEMENT

3.1 EIS/GEM INFOBOOK

Essential Technologies, Inc.[10] developed the EIS/GEM Infobook™, which was adopted by FEMA as the emergency operating software component of the National Emergency Management Information System (NEMIS). Also, it is the software of choice for: Local and State Governments; US Armed Forces including Air Force, Army, Navy, National Guard, and Coast Guard; Chemical, pharmaceutical; utilities; and other companies.

EIS/GEM offers capabilities user can customize the functionalities and user interface of their system to match their organizational needs. It allows users to:

- Create and automate users' own data searches (similar to macros)
- Set up data entry buttons that automatically enter consistent information for users
- Launch third-party software
- Generate special reports
- Display messages and prompts on screen with one click
- Create a collection of searches that run unassisted for automatic briefings
- Search and display data automatically on a regular basis to show the current state-of-the-incident
- Organize just the tasks users use into a single convenient page or spot

EIS/GEM contains hundreds of vital and utilities all with open architecture and in line with industry standards. It is designed as the critical center of users' system-wide information

management, and is used to:

- Interface with additional mapping and GIS products
- Display and interpret spatial output from predictive models
- Take advantage of multi-media enhancements of data
- Launch users' favorite spreadsheet and word processing packages, and Internet browsers
- Read real-time information from data gathering products such as meteorological stations
- Import data from all industry standard databases, saving time and money in data entry

EIS/GEM Infobook serves in almost any computing environment: stand-alone workstations; Local or Wide Area Network configurations; Client/Server architectures (with Oracle or SQL Server); Internet/Intranet/Intra Webs. The system lets users share data, maps, images, and predictive model output between crisis responders in the field and the crisis management coordinator, between branch offices, or between different organizations. InfoComm, built-in message management system, provides users with fast, flexible, reliable connectivity, and serves as the origin and destination for all of users' messaging needs. Basically it distributes messages, taskings, and other data throughout users' LAN with its built-in Message Center. It has capability to extend communication using Internet/Intranet: it can send message to and receive them from external E-mails systems that are MAPI-compliant programs. In addition, InfoComm can integrate EIS/GEM's own ECOMM point-to-point communications program, providing robust connectivity via telephone, cellular, and wireless modems as well as satellite and packet radio.

EIS/GEM improves performance in staff and equipment resource management. It automatically records the action system-wide, noting the deployment in the resource or personnel records by updating available quantities and status. In addition, it enters the

deployment into Operations Log so that users always know where everyone and everything is at any moment. To locate staff and/or equipment, EIS/GEM provides mapping system that is fully integrated with its own multi-level raster maps plus several third-party mapping systems such as ArcView®, MapInfo®, and DeLorme Street Atlas USA™.

EIS/GEM can be integrated with third-party predictive models related to chemicals, waste products, oil or radiation. To predict the impact of various kinds of releases, it is also combined with the latest in sensor technology.

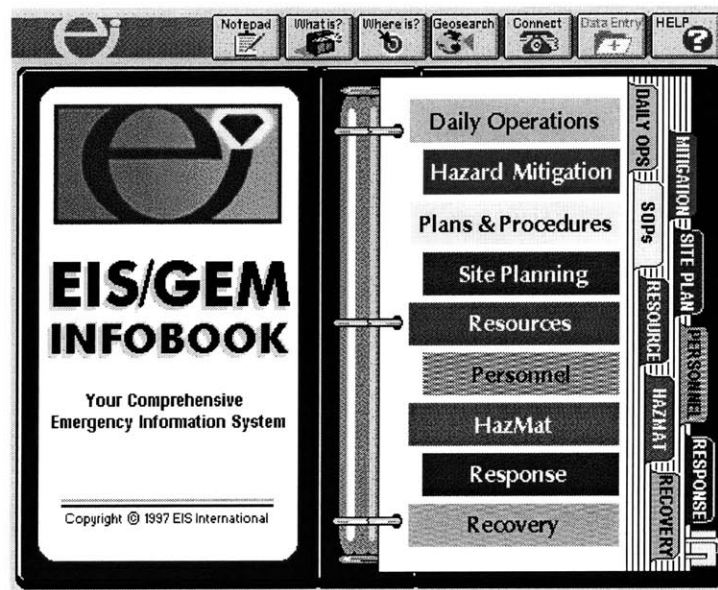


Figure 3.1: EIS/GEM Infobook [10]

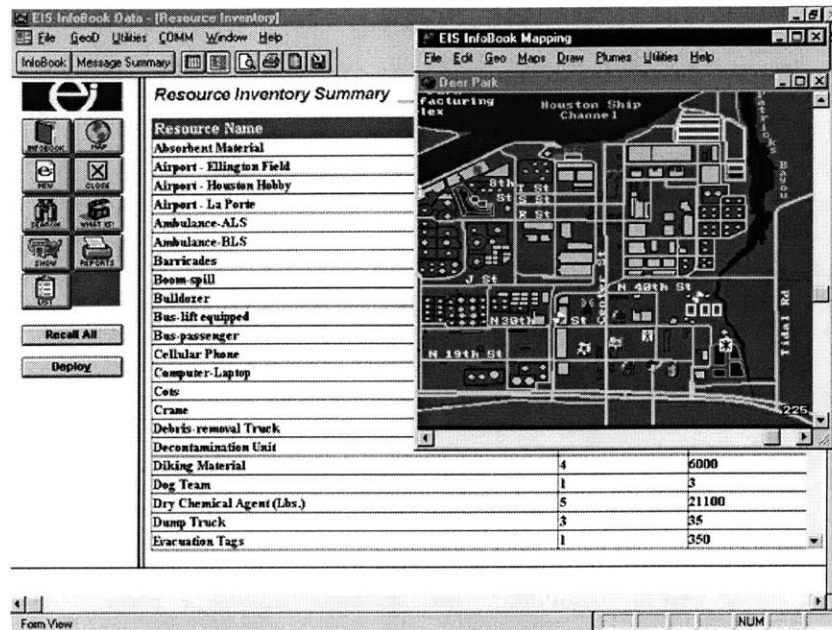


Figure 3.2: EIS/GEM resource inventory [10]

3.2 RESPONSE INFORMATION MANAGEMENT SYSTEM (RIMS)

California's Response Information Management System (RIMS)[11] is a statewide system used to coordinate and manage a State's response to disasters and emergencies. It is a collaborative, web-based system that links the multiple levels of emergency management within the state and automates the state's emergency management system.

The application was developed jointly by the California Office of Emergency Services (OES) and a CSMC[12] project team and has been fielded at over 500 locations with over 2500 active users. RIMS includes several applications including resource management, intelligence reporting, cost accounting and a purchasing system. It has a GIS component to display interactive maps and photos during a disaster.

RIMS is available to all cities, special districts and state agencies within California that have a computer with access to the Internet. Web access to RIMS is controlled by user IDs and passwords. Web browser clients and Internet access are standard and supported in

nearly every government agency, and those agencies that do not have access now soon will. RIMS user base has expanded from 137 agencies to 2500 by developing the internet-based system.

The RIMS applications have been converted to Web accessible format and new, more powerful Lotus Domino Web servers have been installed at each of OES' Regional Emergency Operations Centers. In addition, high-speed Internet access has been added at each OES Region to handle the increased RIMS user traffic.

Clients will still be able to dial in directly to their OES Region using regular phone lines or the Operational Area Satellite Information System (OASIS) (only Operational Areas (OAs) for backup access). Instead of dialing directly into a Lotus Notes server, users will dial into OES just like accessing an Internet Provider. Clients will be able to simultaneously access both RIMS and other OES Lotus Notes applications if they have a notes client. Even though they will be using Web browsers directly to OES for RIMS, they will not be able to get beyond RIMS with their Web browsers.

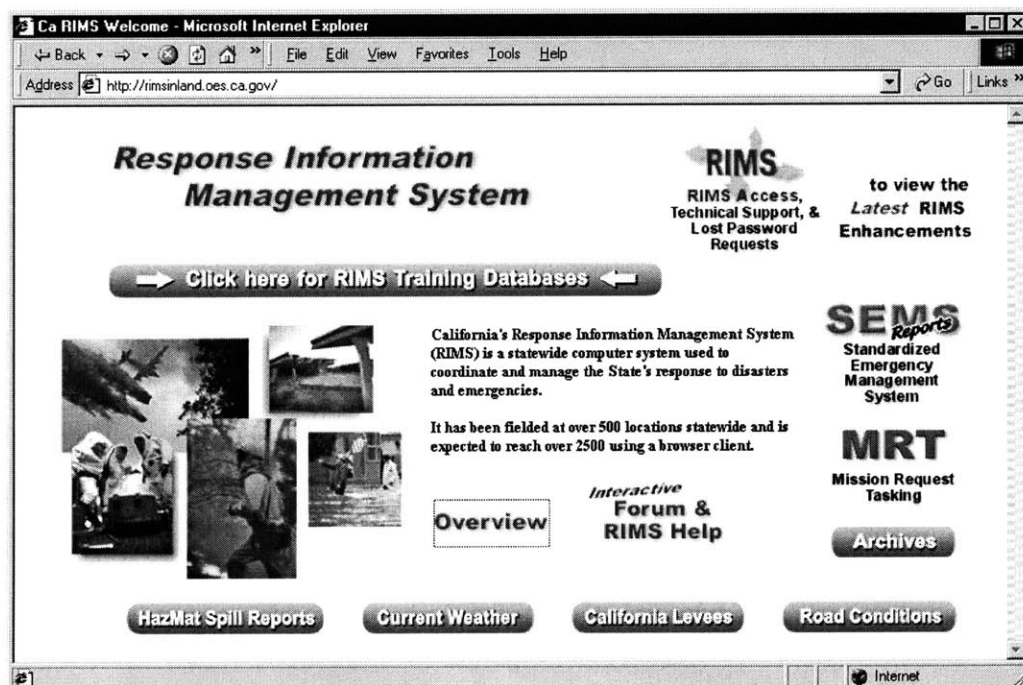


Figure 3.3: California's Response Information Management System [11]

3.3 E TEAM

E Team[13] is an Internet-based workflow management application designed for emergency and event personnel. It allows Web-connected users to enter and access standardized reports, requests, and directives; view and edit annotated maps; and access common databases in order to manage an event or disaster. It is a fully Web-enabled, ODBC and CORBA compliant application, based on Lotus Domino™ web server software, ESRI-based map utilities accommodate vector and raster data.

The solution is comprised of Windows NT servers loaded Lotus Domino, E Team software and users equipped with Internet browser such as Internet Explorer and Netscape. The system enables users to share the disaster-related information such as the status of an incident, where the personnel is deployed, and the location and the availability of resources. Users can enter the information into a set of pre-defined forms, and the information can be cross-referenced to a map of underlying database. IT managers can set policy on which users can access what data.

Bellow is a list of key features and benefits of E Team product.

- ICS compliant: E Team software supports emergency teams using the Incident Command System at all levels of command. Every participant at all command posts and operations centers has access to whatever view of the overall status that they need and are authorized to see.
- Data sharing: E Team allows every user to post and retrieve information to a commonly accessible repository.
- Maps and overlays: Locations of key incidents and facilities are available on map overlays. Clicking on icons navigates users to the source report or data.
- Selectable levels of detail: Managers can see summaries that show what may need

attention. Clicking summary rows brings up the details.

- Security and auditability: Users can be given different levels of access. All transactions are captured and stored automatically.

Functions of E Team support several kinds of information that are vital to emergency management.

- Incident reporting: All ICS-specified information can be quickly filled in on intuitive forms.
- Rescue request tracking: Summary displays are accessible to every user and show the status of requests and who needs to take action.
- Infrastructure status reporting: Simple reports indicate who needs help and who can help. Public facilities managers can easily report and view summary or detail, operational capability, damage and assistance required. Current status of hospitals, medical facilities and shelters can be seen in summary, in detail and on the map. Web-ready release of public information can be quickly prepared, routed through an edit and approval chain, and posted to the Internet.

These are the upcoming enhancements to be implemented to E Team.

- Mobile clients on laptops and Palm PCs connected by cell phone, wireless WAN or radio
- Mark-up map overlays
- Messaging and work group organization tools
- Contingency planning tools and embedded training

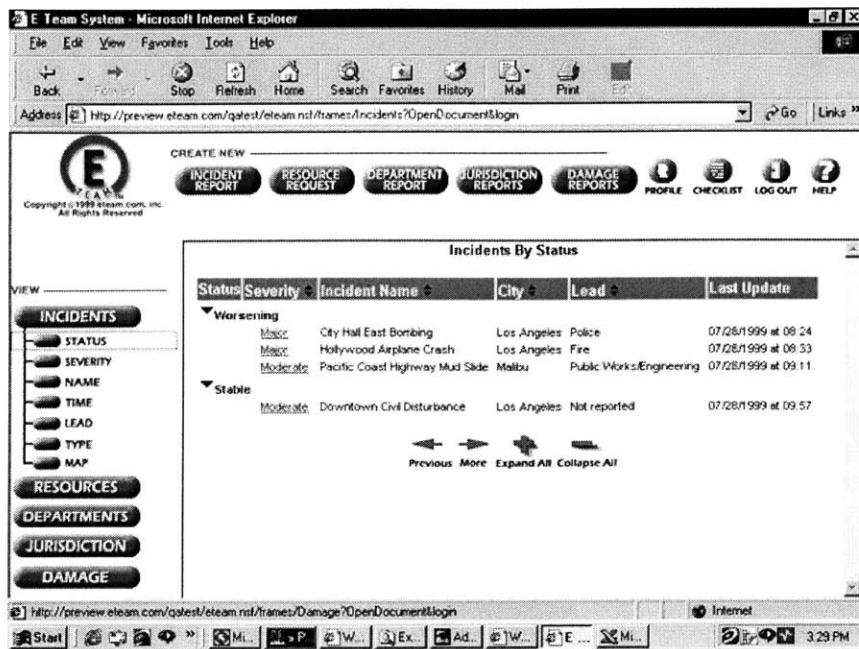


Figure 3.4: E Team Internet-based Emergency Management Software [13]

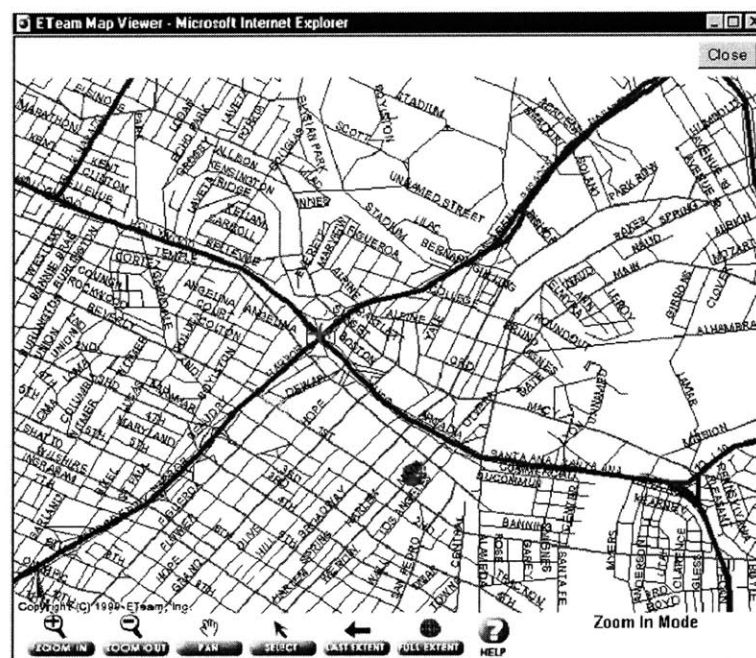


Figure 3.5: Information cross-referenced to a map (E Team) [13]

3.4 LOTUS SOLUTIONS

As mentioned above, California Office of Emergency Services (OES) utilizes Lotus Notes applications, and E Team's solution is based on Lotus Domino. Lotus Notes and Domino are mainly the client/server combination for collaborative environments including Internet messaging and directories, group scheduling, and discussion. Figure 3.6 shows the key Notes and Domino hardware and software components. The hardware components are Notes client computers, Domino server computers, and the network that connects them. The software components reside on the client and server computers. The same three-level software architecture is used on both client and server computers, and each key software component belongs to one of the levels:

- Client and server programs
- Notes Object Services (NOS)
- Databases and files

Lotus Domino servers and Lotus Notes clients together provide an electronic messaging system that is reliable and flexible enough to meet the evolving needs of the collaborative environments. Domino and Notes messaging system supports not only traditional e-mail that relies solely on point-to-point, point-to-multi-point communication based on sending messages from one person to another, but also group discussion that lets groups of users work together on a single instance of a document, or interact with each other on a range of topics. Lotus defines two messaging paradigms: one is many-to-one that denotes many users collaborate on a single topic, and is supported in a Domino Document Library; the other is many-to-many that denotes many users collaborate on a range of related topics and is supported in a Domino Discussion.

A Domino discussion database serves as a meeting place where a workgroup can share ideas conveniently and securely. A Notes client automatically lets discussion participants

trace the sequence of replies in a discussion, using collapsible sections to speed navigations and without the need to wade through irrelevant documents. Documents can be categorized by topic, with views available by category, author or date.

Domino and Notes calendaring/scheduling features provide group scheduling, time management and task management. These features also support development of organizational calendaring applications, such as workforce time management. Customers and Lotus Business Partners can utilize Domino and Organizer APIs, Visual Basic, LotusScript, ODBC, OLE 2.0 and a variety of other methods to access data and automate processes. The Domino data store holds both mail and calendar information. Notes users can benefit from real-time access to free/busy time information. Domino and Notes infrastructure serves as the delivery vehicle for meeting notices and replies, while Domino replication ensures a robust, scalable solution for the mobile or occasionally connected users as well.

Domino provides users with ability to manage document-based workflow. Domino greatly simplifies document management process like:

- Routing documents for review and publication
- Tracking the status of routed documents
- Enterprise and inter-enterprise document distribution
- Providing robust, flexible security for documents and/or components of documents down to the field level
- Making documents accessible either widely or selectively
- Organizing and linking documents
- Document maintenance and archiving

With Domino and Notes Messaging's full text search and query capabilities, along with the Notes clients' customizable views and hierarchical folders, users do not have to remember

file names, authors or dates to retrieve information.

Lotus Domino and Notes messaging system provides users with a variety of collaborative services. These services are not equipped with synchronous features: group discussion allows many users to post messages, but a meeting place is not so organized. A group discussion service is no more than a traditional Bulletin Board System furnished with more security features.

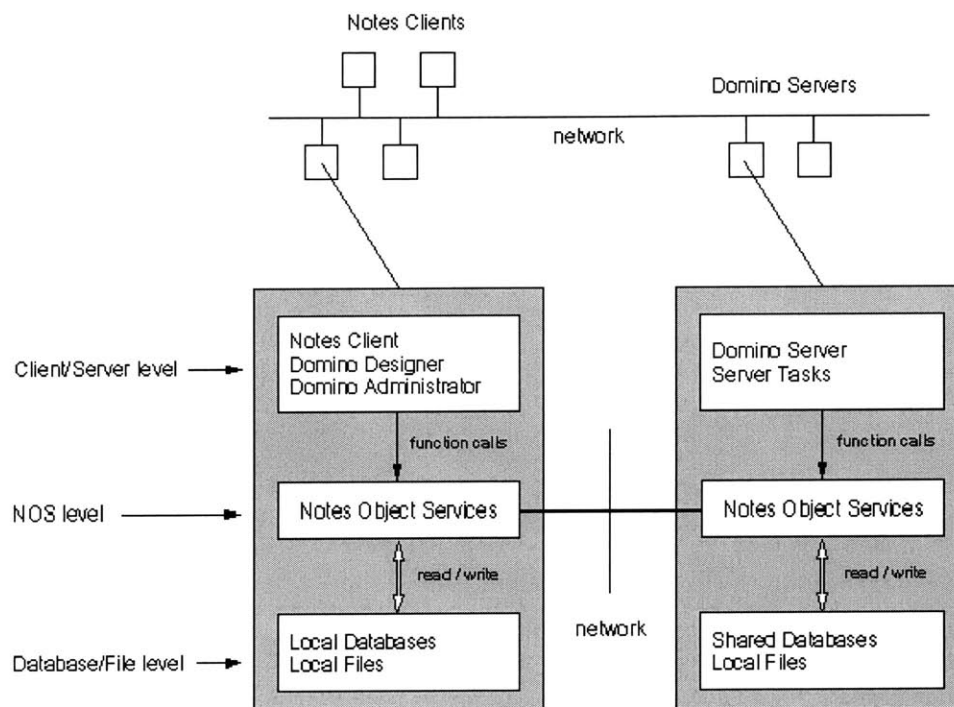


Figure 3.6: Lotus Notes and Domino components [14]

3.4.1 Lotus Domino

Domino's integrated application services optimize the platform for rapid delivery of the collaborative Web applications. Built-in connection services provide live access to leading relational databases, transaction systems and ERP applications. Also users can use leading third party Web development tools along with Domino's integrated development environment.

Domino offers the industry's most comprehensive support for Internet

messaging standards, with Internet addressing, SMTP routing and MIME content support all native. Plus full support for E/SMTP, S/MIME, SSL, POP3, IMAP4, LDAP, HTTP, HTML, SNMP and all the rest. Domino delivers the utmost in interoperability with users' current tools and systems.

Domino runs on several platforms, including Windows NT, OS/2, UNIX, AS/400, Linux and S/390. And Domino supports a wide variety of clients, in addition to the Notes client. Messaging features are available to Web browsers and Internet mail clients (such as POP3 and IMAPv4 clients); directory features are available to browsers and LDAP clients; discussion features are available to browsers and NNTP newsreader clients; and administration features are available to browsers as well as the Notes client. In addition, Domino offers the platform for designing dynamic Web applications, and with the Domino Designer, users can easily build a single application that looks and runs the same for both the Web and Notes.

3.4.2 Lotus Notes

Lotus Notes is an integrated e-mail and collaborative application software for the Internet and corporate intranets. An intuitive, Web-inspired environment, Notes integrates users information sources, including e-mail, calendaring, group scheduling, and to do list. Notes users can exchange messages via the Internet, work with any Web application, read and post topics to Internet newsgroups, search Web directories, and use X.509 certificates for security. Notes lets users access their e-mail and other applications while they are off-line and gives them the option to synchronize their work later.

Tight integration with the Lotus Domino Server makes Notes the most powerful e-business client available, delivering unmatched functionality, security and customization options. Notes enables organizations to maximize the value of their Domino infrastructure – from standards-based messaging to built-in collaborative applications like discussions and document libraries, to the unlimited potential of Domino-based collaborative environment solutions. Together, Notes and Domino enable organizations to lower the total cost of a messaging and collaboration infrastructure, while enhancing users' productivity and reducing

administrative overhead.

3.4.2 Mobile Access to Lotus Services

Mobile Notes, via the Mobile Services for Domino server, provides secure anytime, anywhere interactive access to Notes Mail, calendar and directory services from Web-enabled devices including cell phones, pagers and PDAs.

Together, Mobile Notes and Mobile Services for Domino (MSD) offer organizations:

- **Robust security:** Lotus Notes and Domino represent one of the most secure data networking environments available. MSD builds on these proven capabilities, in combination with new technologies and industry standards like SSL, to address the unique challenges presented by wireless data access. MSD enables organizations to associate an authorized user with each mobile device, track what network a device is used on, encrypt data in transmission, and more.
- **Centralized control:** MSD provides a full complement of wireless administrative capabilities, which are integrated with the Domino Administrator. So you can easily manage all your organization's wireless users and devices. Familiar Domino security features give you total control over who gets into your network, and what gets out over it.
- **Extranet access:** MSD includes a Domino-based Web site that customers and colleagues-inside or outside your organization-can use to contact mobile users via e-mail. The site is easy for administrators to configure and for visitors to use, ensuring both privacy and convenience.
- **Extended reach for applications:** Rapidly create or extend Domino applications, via standard Domino APIs, to automatically send e-mail to mobile phone users in response to events. MSD also enables you to send alerts and notifications to users of one- and two-way pagers and PDAs, via Domino applications.

3.4.2.1 Mobile Services for Domino

Mobile Services for Domino is a server that acts as a proxy, facilitates secure communications, and delivers Domino's leading messaging and collaborative applications to Mobile Notes – the access point that provides users with a Notes-experience from Web-enabled devices including cellular phones, pagers and PDAs. Mobile Services for Domino also provides push technology to deliver paging and short messaging services.

Features new to Mobile Services for Domino server include:

- Unified installation: allows organizations to easily setup Wireless Domino Access and/or Messaging Services using a single installation program.
- Centralized administration: provides centralized administration to configure the server, devices, security, and users. And to record events and manage all the wireless services.
- Messaging service enhancements: have been expanded to include support for EMI/UCP, X.25 and SMS in Europe. Users will also see improved reliability using Service Based Query, and a greatly enhanced console user interface.
- Expanded international support: allows characters sets like UTF-8, Shift-JIS and Korean to be recognized and translated. And international dialing is supported.
- Enhanced stability and performance. Provides Notes and Domino customers with dependable wireless messaging within their enterprises but outside their trusted firewall.
- Stronger security: allows for longer Domino passwords, more stringent security on database access, and end-to-end session logging for easier management and auditing.
- Extended support: provides additional protocol, device support and wireless data network support.

3.4.2.2 Mobile Notes

Lotus Mobile Notes extends the value of Lotus Notes and Domino by providing secure, real-time access to users' e-mail, calendar and directory information from Web-enabled mobile phones. With Mobile Notes users can send and receive Notes mail messages, view users' calendar and schedule appointments from any location. Integration with the corporate Domino directory lets users search for names, address e-mail messages and dial the phone automatically.

3.5 COLLABORATIVE ENVIRONMENT AT MIT

The DaVinci Initiative[15] investigates the use of information technologies to improve online collaborations. The focus of the research at MIT has been in the area of meetings, both online and physical, and how the participants interact, leading to some general observations of collaborations and their participants. Of these observations, the protocols and rules used during a meeting have been the primary focus of research at MIT. Meeting protocols establish how information is shared within the collaboration and how control of the meeting is established.

CAIRO (Collaborative Agent Interaction and Synchronization)[16] was developed as a realization of the collaboration research at MIT. CAIRO is a meeting environment that encompasses many of the research results such as membership and meeting protocol enforcement. Using a strong implementation of meeting protocols, CAIRO uses the Internet to communicate in a highly structured environment that includes protocol-enforcing agents to simulate a physical meeting with a facilitator.

3.5.1 Meeting structure

Within the CAIRO meeting environment, there is a strong sense of membership in a team. In addition to the long-term regular participants, there are transient participants, who are involved for a subset of the entire project. These transient participants include people such as guest speakers or experts. Communication between any participant (regular or transient members) takes place via conversations between all participants or a subset of the participants. Conversations that do not involve all of the meeting participants are

termed whispers, modeled after the act of whispering in a physical meeting. As the size of the meeting grows and the number of issues to be resolved privately increases, breakout meetings are formed to further discuss and resolve key issues among a subset of the team. Breakout meetings are formed to discuss and resolve specific issues, and generally involve a subset of the parent meeting participants. As finer grain detail is expected, outside participants may be asked to join the breakout meeting based on their area of expertise. Opening up meetings to outside participants, such as expert consultants increases the potential for proprietary or private information to be accessed by unauthorized users. Similar to those in the CAIRO system, policies are in place to limit the flow of information between breakout meetings and parent meetings to ensure a consistent level of security between all meetings that are associated with each other. For example, a breakout mission-planning meeting may need access to equipment information, outside participants may join for consultation on equipment but these transient members should not have access to personnel information for the mission being discussed in the parent meeting.

3.5.2 Information Policies

When participants are utilizing many conversations in a single meeting, the collaboration system manages the knowledge and resource transfer within and between them. For instance, when a mission planner is required to communicate privately with a paramedic, the system could enable only the two conversing to access information generated and shared during their interaction, ensuring privacy between the two. The meeting environment is also responsible for making information generated in the primary conversation, such as agendas, drawings, and text-chat dialogs, accessible from the private communications. When a participant enters a breakout meeting, certain aspects of the main meeting data will move with them, and some will remain private to the parent meeting. Due to security issues such as information confidentiality, information policies are established for each meeting based on a set of meeting protocols, or simple rules of conduct.

3.5.3 Meeting Protocols

Within a disaster relief scenario, floor control could be crucial for a smooth and efficient planning meeting. Within any meeting, some level of control is attempted, enforcing

established protocols. For instance, consider a protocol, requiring all participants must ask permission to speak. If a participant chooses to speak without permission they will be breaking the protocols and hence slow the progression of the meeting and potentially divert the direction of the meeting. Protocols usually exist very loosely within a meeting environment because the enforcement of such protocols is left to the meeting participants themselves. When a system, like CAIRO, is deployed, support of the protocols is handled by the application acting as a neutral third party. Protocols make meetings more efficient by acting as enablers, requiring tokens for speech, monitoring speech time and supporting preset agendas, allowing members to keep the meeting on track.

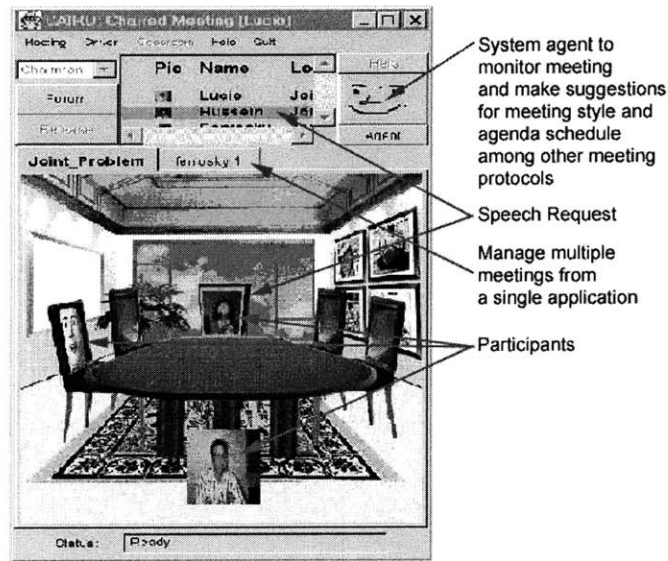


Figure 3.7: CAIRO Meeting Environment [16]

3.6 COLLABORATIVE ENVIRONMENT AT DRAPER

Prior to the partnership with MIT, Draper Laboratory[17] pursued online collaboration by focusing on mission-planning applications. The target deployment at Draper is a military environment that is usually highly unstable and often involves human life. For these reasons, the focus of the research at Draper has led to some unique requirements including persistence of planning data between connections and the ability for agents in the field to

participate in the planning efforts.

3.6.1 Mission Planning Applications

Two of Draper's mission-planning applications were developed to support the needs of modern mission planners, replacing traditional paper-based solutions. They are the Geospatial Application that concentrates on geographical information, such as mapping of regions and routes between these regions and the Temporal Synchronization Application that deals with time constraint issues, such as mode of transportation and travel time. Together these two applications greatly improve the efficiency of the mission planning setting, breaking it into specialized tasks. Both applications were developed independent of any collaboration environment and independent of each other, however both developments used a similar architecture.

3.6.2 Draper Application Architecture and Data Persistence

Collaborative environments typically use applications centered on communication between participants. Once non-communication applications are introduced into this environment, persistence is needed to preserve the application data between sessions. These applications center on data: coordinates of a GIS application or time and actions information of a project management application. Consider a military collaboration environment, among geographically distributed participants using an unreliably network for communication. The persistence of the collaboration data would allow participants to rebuild their session from information stored on the repository rather than lose all access to information generated prior to their connection. To solve both collaboration and repository issues, Draper chose an object oriented database solution, which allowed several instances of an application to attach to the database and, thus, collaboratively work on the same information. Client applications attached to the database and submitted changes to that database. When a change was made on the database, all registered applications were notified of the change. Each instance of the application then displayed the change, resulting in synchronous collaboration. However, performance problems and development difficulties were encountered with this prototype. The speed and reliability of transactions are critical to

effective collaboration environment, but the prototype solution resulted in long update times and therefore unreliable collaborations.

3.6.3 Multiple Device Interface

As Draper research progressed, it became apparent that many collaborators in a mission-planning environment would be in the field executing the plan as it was being developed. It is impractical to expect participants in the field to have capabilities similar to the planning facilities, and isolation of these participants on a separate system, such as two-way radios, would leave them at a great disadvantage and decrease their ability to contribute to the collaboration. Therefore, a goal of the mission-planning objective at Draper is to incorporate multiple hardware devices such as handheld computers and wireless web-enabled devices into the collaboration. Using handheld devices and specialized content based on limitations, field rescuers could stay in sync with mission planning officials, resulting in an efficient collaboration.

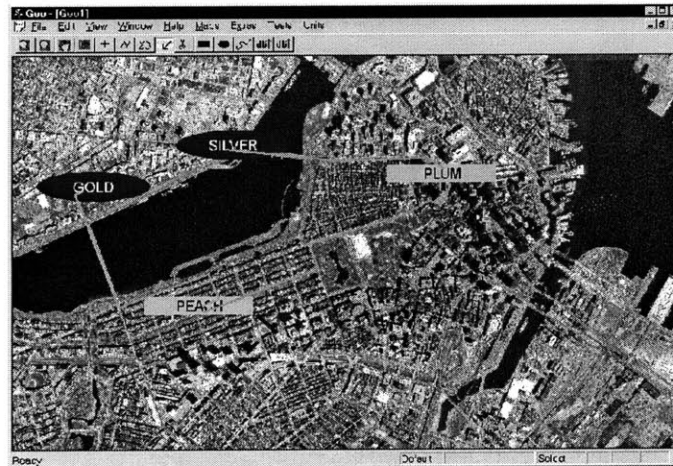


Figure 3.8: Geospatial Application [17]

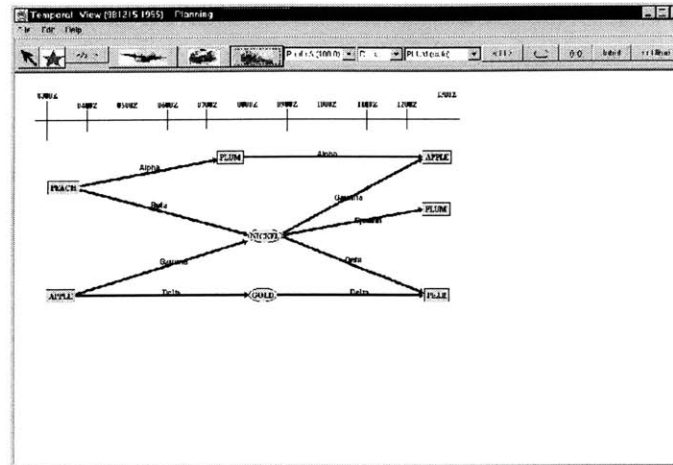


Figure 3.9: Temporal Synchronization Application [17]

3.7 COLLABORATION BETWEEN MIT AND DRAPER

Draper, recognizing MIT's collaboration experience and desiring to improve their application-centered approach to collaboration, established a joint research initiative in the summer of 1998. At the end of the first year, the Temporal Synchronization Application was integrated into the CAIRO system, making use of the collaborative environment. Collaborators could log into the CAIRO meeting environment, having access to the Temporal application as well as existing ones such as project management, utilizing an existing set of protocols. Continuing development of both the CAIRO meeting environment and the Draper applications focuses on system architecture for collaboration, integrating a data repository, collaboration protocols and interfaces to support alternative hardware devices. Joint efforts began and maintain a high level of monitoring of the current state of the art.

4. EMERGENCY MANAGEMENT SYSTEM DEVELOPMENT

4.1 SCENARIO

4.1.1 Participants

When a large-scale disaster happens, a disaster relief mission is performed by teams distributed geographically. The participants are comprised of various individuals and organizations with different expertise. Table 3.1 defines disaster relief participants and categorizes these participants in two ways. First, they are divided by their organizations:

- Search and rescue (SAR) groups that are responsible for locating, extricating and providing initial medical treatment to victims trapped in a collapsed structures;
- Firefighting groups that are responsible for detecting and suppressing fires;
- Medical service group that are responsible for providing assistance for medical care needs;
- Hazardous material groups that are responsible for dealing with actual or potential releases of hazardous materials such as oil, gas and radiation;
- Local/State/Federal office of emergency management that is responsible for gathering information about the disaster area and making decisions about deployment of task forces and requesting assistance from other organization;
- Structural engineering specialist who is responsible for analyzing the current situation where field agents are working on and making their activities safe by providing technical information about structural integrity of the building and the risk of secondary collapses;
- Utilities (electric power, gas, water) that are responsible for analyzing the damage of

their facilities and providing technical information to support field agents' activities.

Also these participants are categorized in terms of the roles described by columns of Table 4.1:

- Field agents that are members of task forces deployed in the disaster area and perform disaster relief activities;
- Team leaders that direct and supervise their team members;
- Coordinators that distribute and organize task forces including staff and equipment;
- Decision makers that collect information, analyze the current situation and make decision to deploy task forces;
- Specialists and advisors that analyze special situations related to their technical background and provide assistance for other participants.

During the mission, all the participants need to be connected to the information platform that allows them to share information and application, and communicate with one another. Table 4.2 shows client devices for each type of participants.

Table 4.1: Participants of disaster relief mission

	Field agent	Team leader	Coordinator	Decision maker	Specialist, Advisor
Search and rescue group	x	x	x		
Firefighting group	x	x	x		
Medical service group	x	x	x		
Hazardous material group	x	x	x		
Local/State/Federal office of emergency management			x	x	
Structural engineering specialist					x
Utilities (electric power, gas, water)					x

Table 4.2: Client device

Personnel	Equipment/Tools
Field agent	Hand Held Device (Palm, PDA), Java-enable phone, pager, Robots (MINDSTORMS)
Team leader	Portable Device (Windows 9x/CE)
Coodinator, Dicision maker	Desktop PC (Windows 9x/NT) / Workstation(UNIX)
Specialist, Advisor	Desktop PC (Windows 9x/NT)

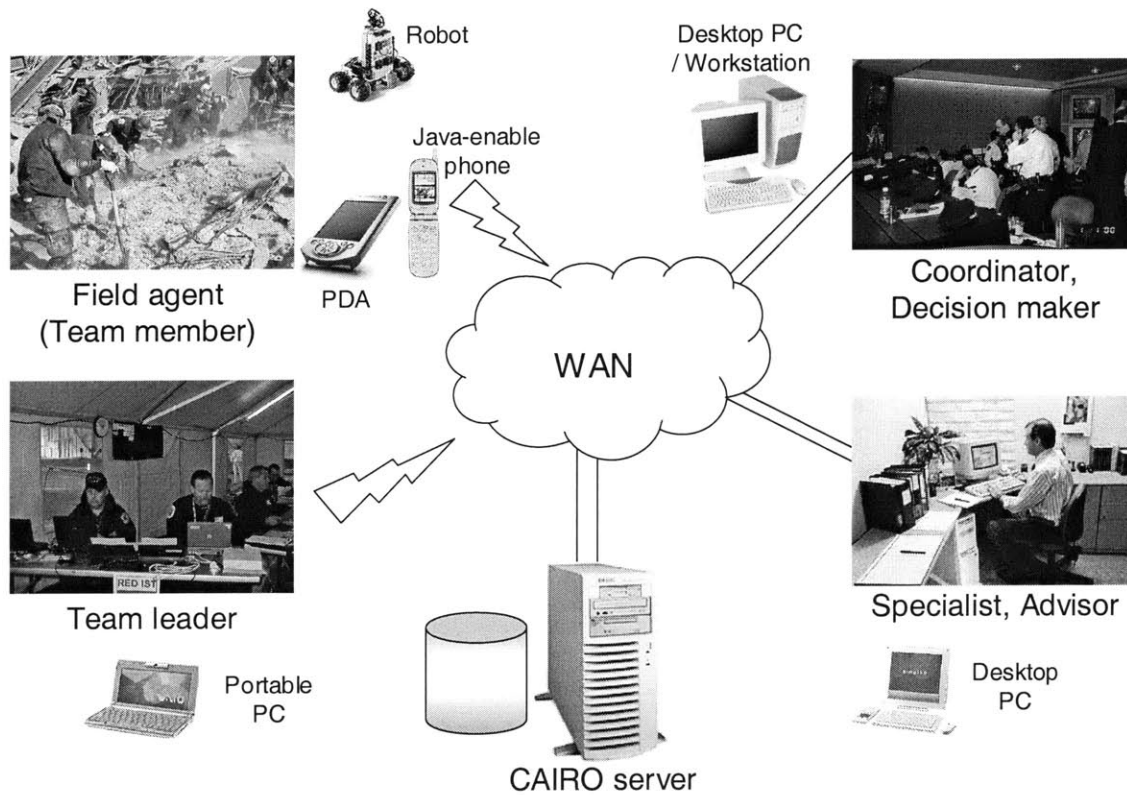


Figure 4.1: Disaster relief mission scenario

4.1.2 Role Description

This subsection describes the activities that each participant performs and interaction between them during a disaster relief mission.

- Field agent:
 - arrives at a disaster area directed by the team leader and the coordinator
 - establishes connection to the collaborative information platform using a hand held device or a Java-enable phone/pager
 - gets instruction from the team leader

- performs disaster relief missions according to the instructions from the team leader
- manipulate the robot to move around and sense surroundings, and monitors feedback information from the robot
- sends feedback and requests to the team leader and the coordinator
- Team leader:
 - arrives at a disaster area designated by the coordinator with the field agents
 - establishes connection to the collaborative information platform
 - receives instruction from the coordinator
 - allocates and gives instruction to the field agents according to the instruction from the coordinator
 - monitors field agents' activities and receives feedback from the field agents
 - sends feedback to the coordinator
- Coordinator:
 - deploys task forces (field agents and team leaders) into a disaster area and gives instruction to team leaders
 - holds connection to the collaborative information platform
 - monitors task forces' activities and receives feedback from team leaders
 - send feedback to the decision maker
 - receive the decision about resource deployment from the decision maker
 - send instruction to team leader

- Decision maker:
 - holds connection to the collaborative information platform
 - monitors all the participants activities and receives feedback from them
 - analyzes the current situation using feedback from the participants and the information stored in the data repository such as GIS data and CAD files and make decisions
 - requests specialists and advisors to join the mission and provide technical assistance to the participants
 - sends instruction to the coordinator
- Specialist/advisor:
 - receives the request to join the mission from the decision maker
 - establishes connection to the collaborative information platform
 - analyzes the current situation using feedback from the participants and the information stored in the data repository
 - monitors the participants activities, especially task forces' activities
 - gives participants suggestions and special instruction related to technical matters

4.2 SYSTEM OVERVIEW

4.2.1 Device and Network Limitation

Field personnel and mission control for planning and rescue missions such as earthquake disaster relief efforts may be forced to depend on radio or wireless communications technology to remain in contact with one another. Field agents typically only had the

device used for communication while mission control personnel had access to more sophisticated equipment. Instructions and guidance from mission control and feedback from the field is critical to keep everyone informed and complete a successful mission, but is limited by the devices used by field agents. A small device could serve multiple purposes, providing a communication link to the mission control and enable field workers access to relief related information such as maps and equipment operating instructions. Devices such as PDAs and Java-enabled cellular phones as well as more specialized devices, have unique constraints on the type and amount of information that can be displayed. Small devices, with specialized content would enable field agents to accurately and effectively participate in a collaboration session, transmitting information to the session and receiving from the session, typically in the form of directives.

4.2.2 System Overview Description

The goal of the emergency management system is to provide a collaborative information platform where the collaborators can share information and application related to disaster relief missions, and communicate with one another so that they can make decisions as easy, fast and inexpensive as possible. The key features of the system are follows:

- Collaborative environment
- Shared information and application
- Thin client
- Multiple device access
- Scalability
- Reliability
- Security

The collaborative environment controls interaction between participants in order to

make decisions effectively. CAIRO (Collaborative Agent Interaction and SynchroniZation) system provides collaborative environment.

The system allows collaborators to share information stored in the data repository and applications in order to promote consistent interactions between participants.

Thin client and multiple device access enable collaborators to use various client devices. Handheld devices, PDAs, phones or pagers are appropriate for field agents since they cannot carry laptop PCs due to the nature of their task.

The system needs to be scalable since a disaster relief mission varies in the scale: it may require mission participants within a city or a state; it may require participants from all over the world.

4.3 USE CASE

4.3.1 Glossary

- **Collaborator:** This term indicates all personnel who are working on a disaster relief mission and have access to the collaborative environment. It includes field agent, team leader, coordinator, decision maker and specialist/advisor mentioned in Section 3.1.
- **FieldAgent:** This term indicates the same personnel as field agent mentioned in Section 4.1.
- **SessionParticipant:** This term indicates personnel who are taking part in a meeting and their actions are organized by a Chair. It includes field agent, team leader and coordinator mentioned in Section 4.1.
- **Chair:** This term indicates personnel who have a privilege to convene and organize a meeting. It includes team leader, coordinator and decision maker mentioned in

Section 4.1.

- Specialist/Advisor: This term indicates the same personnel as specialist/advisor mentioned in Section 4.1. The personnel can take part in a meeting and give instructions and/or advice.

4.3.2 Diagram

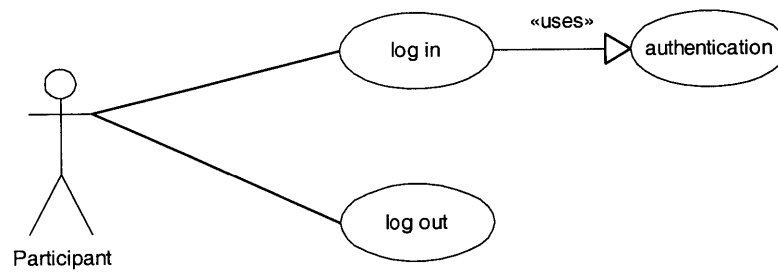


Figure 4.2: Collaborator use case diagram

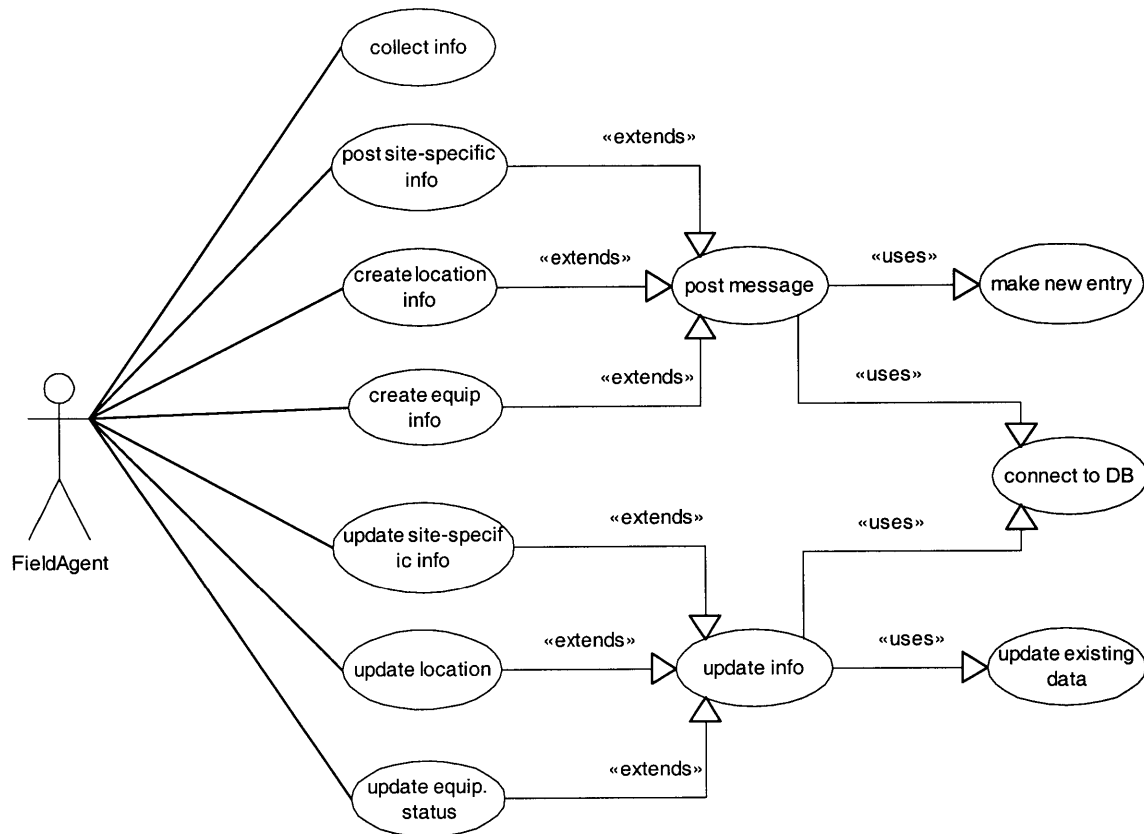


Figure 4.3: Field agent use case diagram

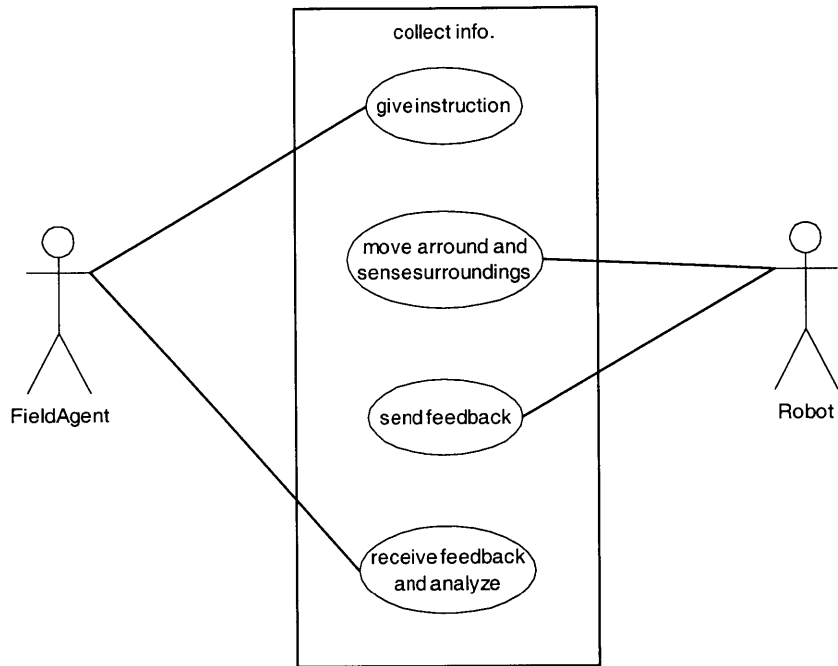


Figure 4.4: Field agent use case diagram (collect info.)

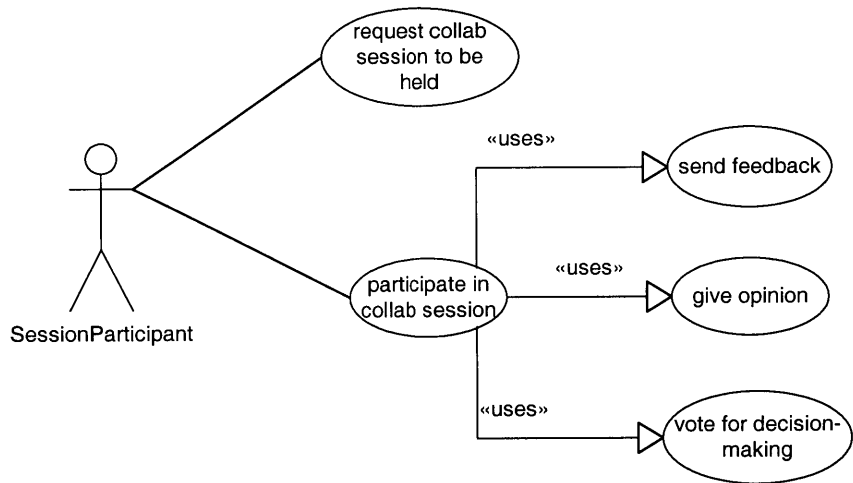


Figure 4.5: Session participant use case diagram

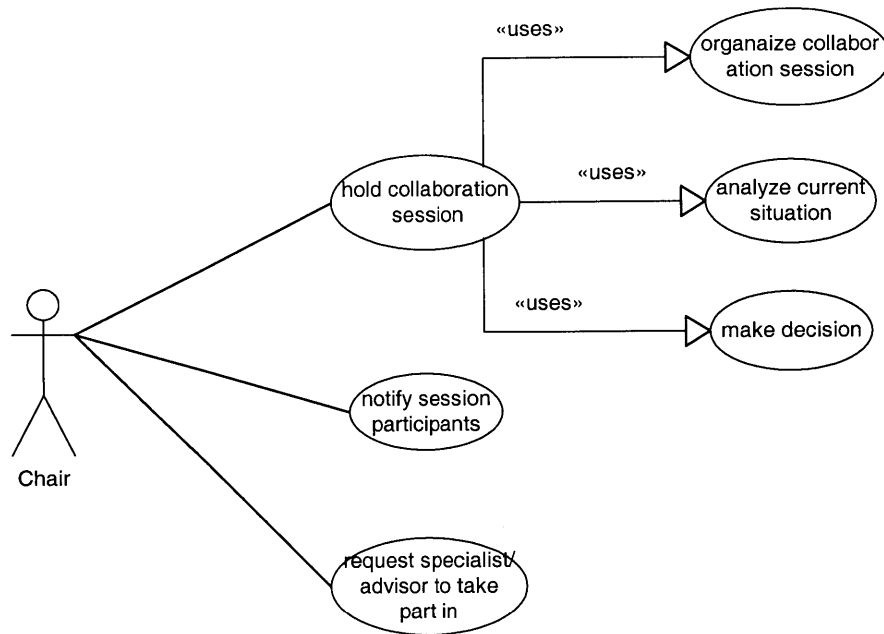


Figure 4.6: Collaboration session chairperson use case diagram

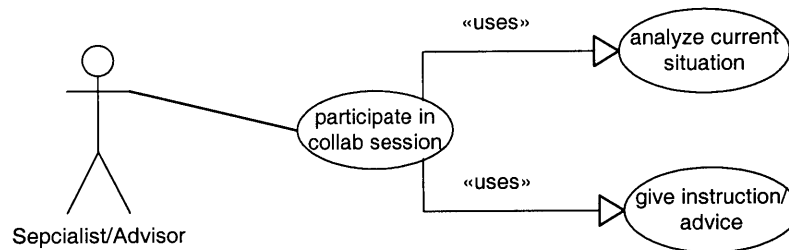


Figure 4.7: Specialist/advisor use case diagram

4.4 COLLABORATIVE ENVIRONMENT DATA FLOW

The data flow of the collaborative environment is described in the Figure4.8. During a disaster relief mission, the activities in terms information exchange are:

- A field agent collects site-specific information, and keeps updating his/her location

and equipment availability.

- A team leader holds a collaboration session with the field agents, receives feed back from them, and gives them instructions.
- A coordinator holds a collaboration session with team leaders, receives feed back from them, and gives them instructions.
- A decision maker holds a collaboration session with coordinators, receives feed back from them, and gives them instructions.
- A specialist/advisor participates in the collaboration session when required, analyzes the situation, and gives other participants some technical advice/instruction.
- In a collaboration session, session participants refer the data repository.

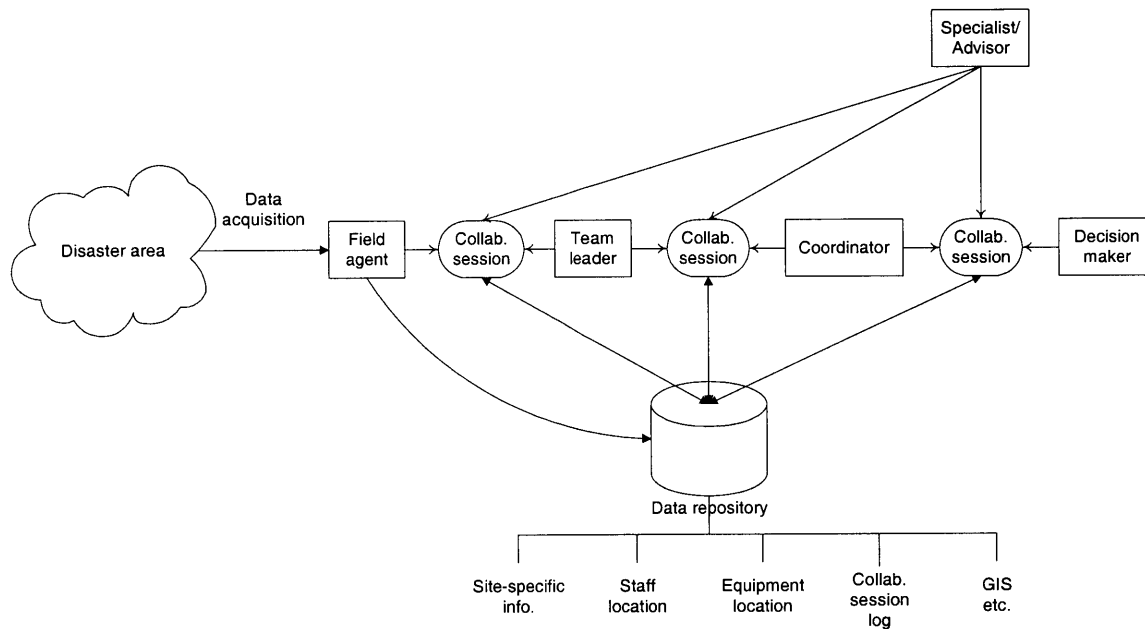


Figure 4.8: Collaborative environment data flow

In a disaster relief mission, a large number of collaboration sessions dealing with different issues are held simultaneously. Since some sessions are often discussing similar issues or topics, interaction between sessions is required as well as interaction between session participants. This study proposes a hierarchy of sessions: several levels of sessions are defined. A session of a certain level is supervised by one of the higher-level sessions, and supervises the lower-level sessions.

Figure 4.9 shows the hierarchy of the collaboration sessions. A decision manager holds the top-level session with coordinators and organizes the session. A coordinator holds the second-level session with team leaders and organizes the session. A team leader holds the third-level session with field agents and organizes the session. A specialist/advisor can take part in any level session. For example, a team leader attends the second-level session, and receives instruction from the coordinator. The leader holds a third-level session and distributes the instructions to field agent. During a third-level session, if a team leader finds the issue is too difficult to solve, the leader requests the coordinator to hold the second-level session or requests the specialist/advisor to take part in the third-level session.

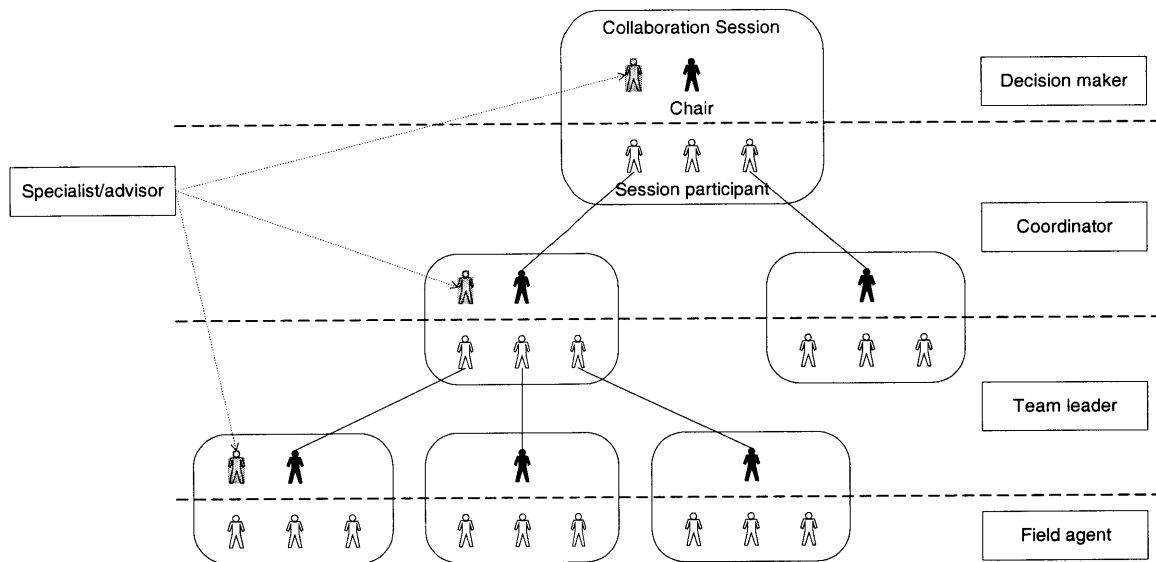


Figure 4.9: Hierarchy of collaboration sessions

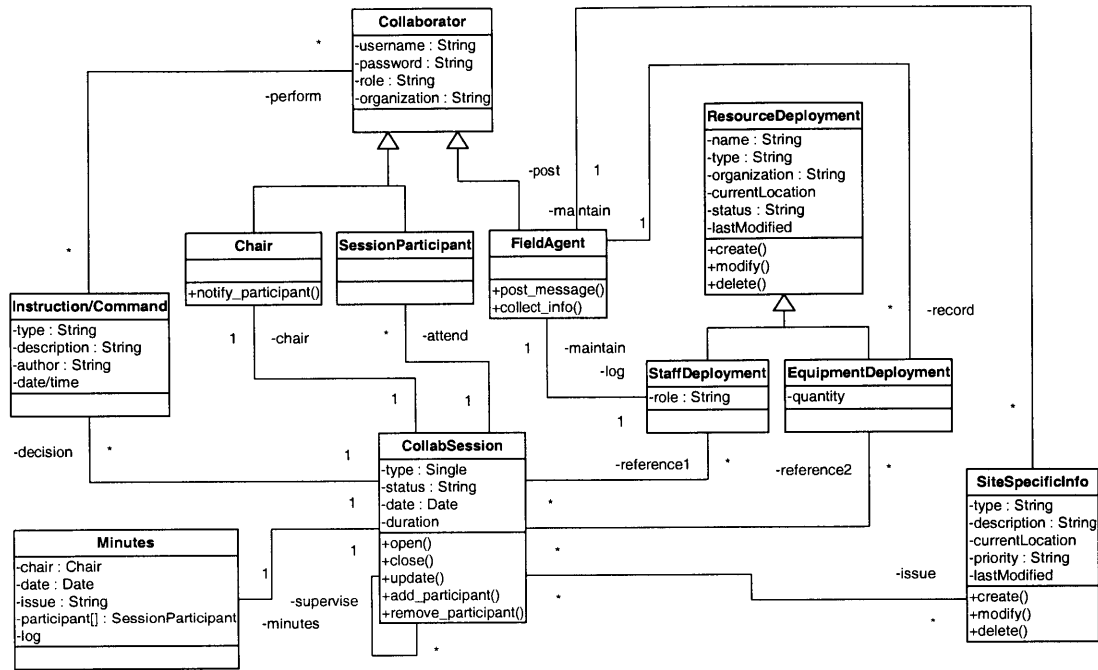


Figure 4.10: Class diagram

4.5 COLLABORATION SYSTEM REQUIREMENTS

Based on the presented use cases, requirements for the collaborative environment can now be detailed. The system will need to provide a structured environment for each collaborative session and information about who the collaborators are. An information policy will be established to determine how information flows and the proper channels for communication. Persistence of collaboration data will be necessary as well as the need for multiple hardware devices to access the system.

4.5.1 Structured Collaborative Environment

As presented, the SAR operation for a large-scale earthquake involved several parties in a distributed environment where multiple types of collaborations will take place using various devices. To solve the problems of confusion and disorganization among collaboration participants, they are placed in a structured environment where policies can be set and

enforced to avoid chaos and/or anarchy. By guiding how the participants interact, the system can help in maintaining control of the collaboration, allowing the leaders of the session to be more effective in coordinating resources. Several key components, which are defined below make up this structured collaborative environment.

4.5.1.1 Collaboration Session

A collaboration session is any grouping of participants for the purpose of working and communicating with one another. Sessions take the form of meetings that involve presentations or intense planning where ideas are developed and explored. The session can last for any length of time and can also be stopped and resumed at a later date if the session has not concluded. For an example of a collaboration session within the scope of the situation presented, consider the search and rescue for a particular building. Within this session there will be several players: the coordinators who are responsible for distributing the resources, the rescue team leaders who are responsible for directing the field agents and in some cases the field agents themselves, providing feedback of their current situation. Joining the session will also be various others such as engineering specialists needing access to the information generated to safeguard the lives of rescuers in case the building may not be structurally stable.

4.5.1.2 Session Organization and Side Sessions

A collaboration session may exist within a complex arrangement of other sessions. Sessions may exist in a hierarchical fashion, resembling an organization structure as in Figure 4.11. The building rescue session presented previously is a single session in the overall organization of collaborations taking place. Participants of the session report one level up to the rescue team coordinator session that is responsible for coordinating all rescue operations. Collaboration sessions may spawn side sessions that are separate, with their own rules and resources, but still belong to the main collaboration session and retain some access to its resources.

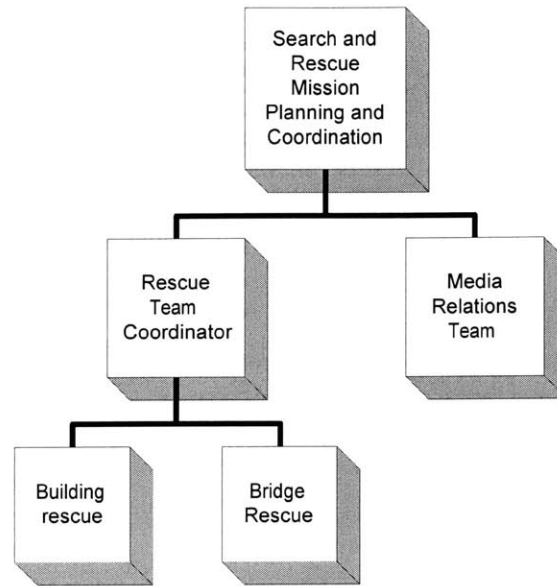


Figure 4.11: Sample session organization structure

4.5.1.3 Conversations and Side Conversations

A conversation is any communication between collaboration participants. The conversation is the channel through which information is conveyed to others. Rather than transmit information to specific participants in a session, information is injected into a conversation that contains the recipients. Whispers, or side conversations are a particular kind of conversation that contains only a subset of the participants involved in the entire collaborative effort. For example, consider a building rescue session that has many participants. Communication between all collaborators will certainly take place, sharing information and directing resources. Additionally, some of the collaborators may have questions concerning an order. These cases can be handled in side conversations, involving only the parties that are necessary, not interfering with everyone in the main session.

4.5.1.4 Protocols

Protocols are the rules of collaboration sessions, their organization and the conversations within each session. Protocols limit the flow of information, control the

number of participants speaking at the same time and maintain a level of control. The protocols of a meeting define the process by which a participant is able to communicate with others often requiring permission of a session leader. A sample flow diagram of how protocols handle a participant sending a request to speak is illustrated in Figure 4.12. Protocols act at all levels, from individual participant requests to side session requests for access to main session information. For example, consider two participants attempting to communicate and neither one is in control of the meeting. If the meeting is setup to disallow such conversation, the chairperson will be alerted. If approved, the two participants are allowed to enter a side conversation; otherwise they must communicate using the session wide conversation involving everyone. Protocols are effective in situations involving participants that do not know each other. In a search and rescue operation or other quick formation collaboration, there is no time to build relationships, forcing participants who have never met to immediately trust each other and recognize orders given by supervisors. By forcing the participants to abide by rules, authority can be established and maintained, allowing a single participant to control a collaborative session involving participants they have no previous relationship with.

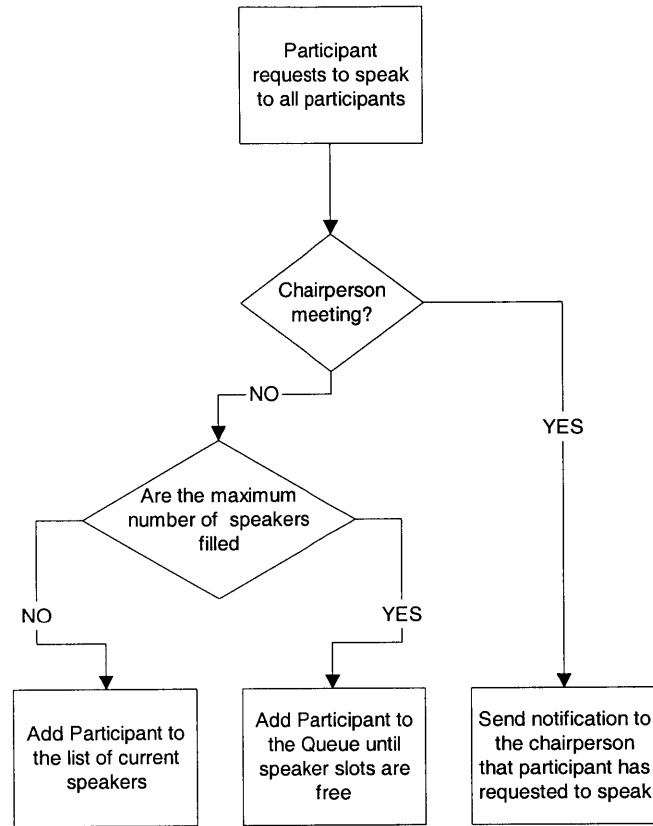


Figure 4.12: Protocol handling of speech request

4.5.2 Representation of Collaborators

Within any collaboration environment, membership is necessary, enabling individual participants to know who is talking and who is receiving or providing information within an often-faceless collaboration. Collaboration sessions involving participants that have never met introduce issues of trust and acceptance. Visual representation and contextual background information help participants understand each person's position in the session and their credentials on an as-needed basis. Knowing who is speaking and their qualifications greatly improves interactions among participants, which justifies protocol enforcement and establishment of information policies. For example, in rescue operation management, the ability to visually see when field agents are participating in a meeting, and access their background information is crucial in making resource management and allocation decisions.

4.5.3 Abstracted Information Policy and Protocol Enforcement

Between sessions and conversations, knowledge and resources need to be transferred to ensure current information is being used. A field rescue collaboration session is at a great disadvantage if it cannot access information generated in a resource management session that is responsible for directing field agents. In a hierarchical, multi-session environment, information dissemination and encapsulation are of key importance. During a single collaboration session, an unknown number of applications are used, requiring the information policy to be independent of any application. The information policy can be implemented in a central location, acting as the interface between the participants and the collaboration session. This location will ensure the policy is secure and can be upgraded on a system-wide basis. By removing the policy implementation from the client, control can be centralized as seen in Figure 4.13. For example, a client application can join the session, abiding by the existing rules without having to know what they are or how they work.

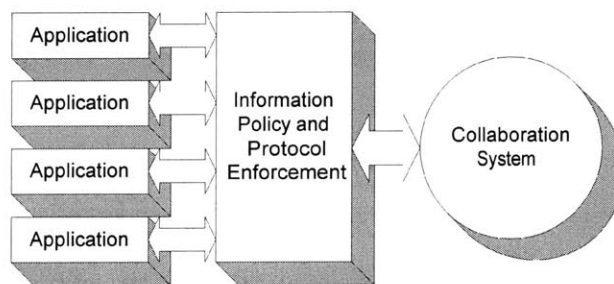


Figure 4.13: Placement of information policy

4.5.4 Persistence of Collaboration Data

Information generated during one session needs to be available to other sessions, during and after. Persistent storage of all information generated during a collaboration session maintains consistency among participants and can be used later for detailed inspection of collaboration sessions. Training exercises and facility updates directly benefit from the examination of historical information, placing an additional requirement for persistency of collaboration information.

4.5.4.1 Consistent Representation Between Collaborators

Utilizing a common repository for all transactions, collaborators can be assured that their session is up to date with others. Each transaction affects a single source, enabling latecomers to build the session to the current state or retrieve the current state of the session in a single request. During loss of service, participants can store transactions locally, executing them once a connection to the session is re-established. For instance consider a case where mission-control participants are guiding rescuers through a building, sharing a blueprint. The field agents will lose network connections frequently, following which they will rebuild their session from information on the repository, as though they never left. For instance, if a directive was drawn on the blueprint indicating where to proceed next while the field agents were offline, when they returned, they would retrieve that directive from the repository and know what their next direction would be.

4.5.4.2 Session Replay and Analysis

Since all transactions are executed on a repository of information, the transactions themselves can be stored. Transaction logs can be replayed at a later date for study of a collaboration session or to search for a particular piece of information. Replay features are very important for training sessions and can be invaluable for studying the performance of participants in a particular collaboration session. High-level collaboration sessions can also study lower sessions in real time, monitoring progress and providing feedback. For instance, rescue training may involve monitoring a past collaboration session to study the decisions made and how successful the execution was. In depth study of the session information will lead to improvements in the environment to work in the widest array of situations.

4.6 SYSTEM ARCHITECTURE

With detailed requirements in place, a robust, scalable and portable solution can be detailed. The best way to accommodate various configurations and portability is to split the system into modular components that integrate to form the system. Thus, the design for the collaboration system is broken into components performing individual tasks such as storage of session information or managing the collaboration. The components can be deployed in

various configurations, adapting to the current demands of the collaboration and the available resources. The system design incorporates the implementation of a protocol enforcement engine, a data repository, the collaboration engine and client application requirements. To address the needs of different collaborative applications, the system is designed to be scalable and portable, working in varying conditions and able to change with future developments and further research. The following section presents the design of the collaboration system components and the technologies behind each of them.

A system to support collaboration and collaborative applications in a high stress situation such as disaster relief must be robust, flexible and portable. Thus, the design of the system developed in the research effort presented in this paper is based on a component-based architecture. The system would be able to operate under various conditions and will accommodate new technologies. Each component can be redesigned later with little impact to the other components. Relying on components to complete key tasks enables the system to de-couple the processes required for an effective collaboration such as data storage and protocol enforcement.

The system architecture is based on the Model-View-Controller architecture shown in Figure 4.14. Using this architecture, the data component, or model is separated from the user interface (view and controller). The controller represents the channel through which the user interacts with the model. For example, consider a database with GIS latitude and longitude information representing the model component. The view may be a map or a text field that displays the coordinates or plots zones based on the arrangements of the points in the database. The controller would be the interface through which clients entered new points and manipulated existing ones in different types of user interfaces. The MVC architecture is based on individual components that perform specific functionality independently, resulting in a more robust and scalable solution.

In this architecture, to fulfill the presented requirements, four components with targeted functionality are necessary to give the collaboration system maximum flexibility and scalability. The four components are: the client application, the information policy, the data repository and the collaboration manager (see Figure 4.15). Separating each component, one

can be modified with limited effect to the others. Additionally, the interconnection of system components is flexible so when one component fails, another instance of that component can be substituted with little effort.

The client application is the interface for the collaborators to the system such as a CAD tool or a collaborative project management application. The information policy serves as the filter for transactions made by the client applications, enforcing protocols at all levels to everyone in the session. The data repository is responsible for storage of collaboration data, such as who is in the session and any application specific data such as what is being drawn on the application. The collaboration manager drives the synchronicity of the system, connecting client applications. The collaboration manager is designed as a separate component to allow sessions the option of using it or not, resulting in synchronous and asynchronous respectively. Separation of real-time (synchronous) from asynchronous collaboration also enables the system to deliver content to various devices, as some will have limitations that prohibit synchronous communication.

Each component is defined as an abstract specification allowing custom implementations to work within the system. This is important for flexibility, as no system can be designed to handle all possible cases. As a new application is introduced, or new functionality is needed, new components can be developed for a particular implementation.

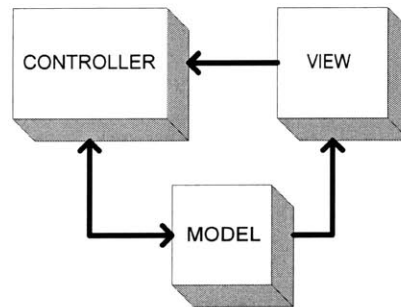


Figure 4.14: Model-view-controller architecture

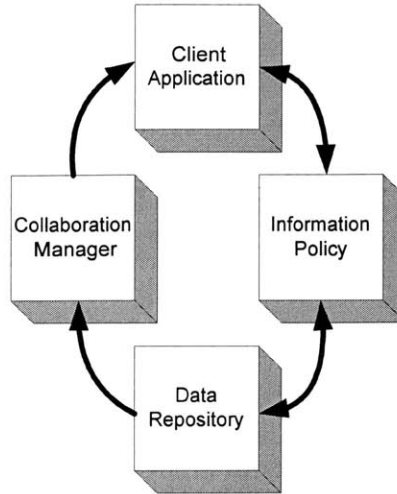


Figure 4.15: Four components of the collaboration system architecture

4.6.1 Client Application

The client application is designed to be open and flexible, accommodating new and existing applications. Requirements placed on applications are minimum, allowing legacy applications to be incorporated into a collaborative environment quickly and easily. Two basic requirements are placed on client applications, both of which are optional: the input to the collaboration system and the output. The application can be designed so it can function properly in any configuration: stand-alone mode without any collaboration, input only, output only or both input and output. A sample configuration of a client application is shown in Figure 4.16.

The input component is the hook into the information policy component that controls the collaborator's interaction with the system. Through this interface, the client sends all transactions to the repository (changes to the application data) and requests permission changes. Responses to requests and transactions are fed back into the input component, alerting the client of errors or rejected requests.

When changes are made to the repository, the collaboration system sends notifications to the collaborators via the output component of the client application. Registering with the collaboration manager, the output component is notified when information changes within the confines of the collaboration session. Once notified, the client application must then

decide what to do with the information.

The exact path for a single transaction depends on the desired functionality and implementation of the application. By design, a request first enters the repository and then notification is sent to all clients, including the client that made the transaction (Figure 4.17B). Using this circular path, the client application is never made to believe that the transaction took place until the system sends notification. Following the path shown in Figure 4.17A, the local copy of the application data is updated concurrently with the collaboration data. However, a problem may be encountered if the transaction does not complete the requested operation and a notification is not received, because the client must then undo the changes to the local data, placing additional burden on the client application. However, this approach makes sure that all clients are synchronized throughout the session.

As mentioned, each application can implement one or both of the components described above. Examples of clients that implement the input component are data entry applications: clients that enter information into the collaboration session but need no notification when others enter information. Such clients may be agents collecting information in the field. Synchronous participation in the collaboration is not a requirement for these clients, however the information gathered might be useful to others in the session that are implementing the output component of the client application. Other applications implement only the output component, for example a “view” type application. These applications may be in the form of mapping applications, displaying locations of field agents on a projection screen. No input to the collaboration session is allowed through the display device, but access to the information and the powerful display capabilities are necessary to collaborators.

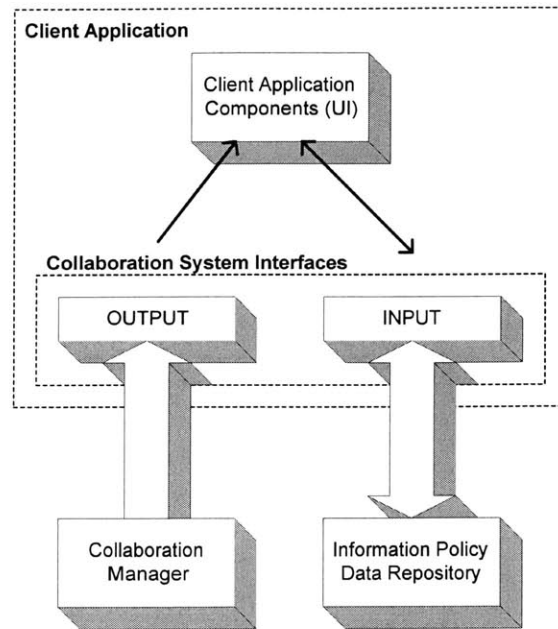
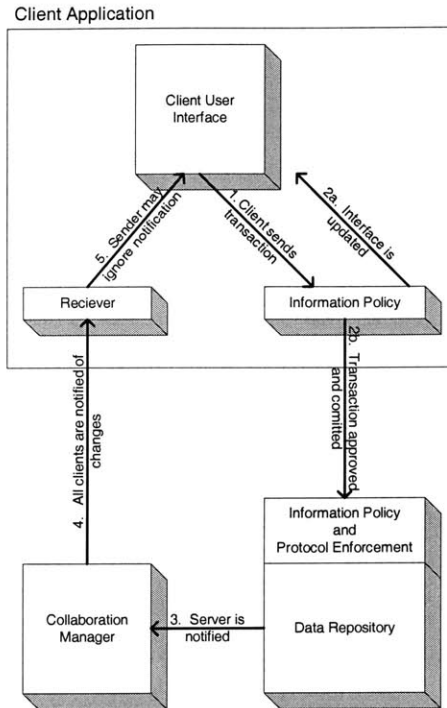
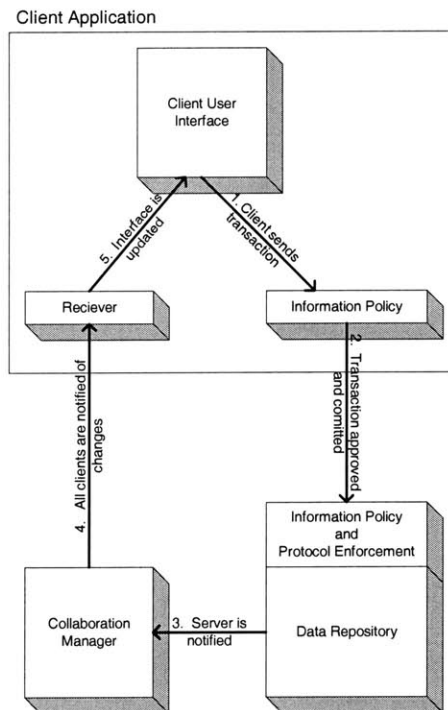


Figure 4.16: Sample client application configuration



A. This approach does not take advantage of the persistence capabilities of the repository and must rely on the local implementation of the information policy to ensure the transaction will commit. Thus, the Reciever component ignores any message it receives that was generated by the client to which it belongs and only accepts messages from other clients.



B. This approach takes full advantage of all components. The architecture presented is similar to that of a distributed Model-View-Controller.

Figure 4.17: Transaction paths for client application

4.6.2 Information Policy

The information policy component enforces the protocols chosen for the collaboration session. Every collaboration session operates under a specified set of protocols that are changeable at any time during the session. Protocols are determined through parameters such as number of speakers, how long someone can hold the floor and whether or not the session has a chairperson; existing configurations for protocols have been developed at MIT. The information policy component enforces access to information through the parameters of such protocols. The information policy is not responsible for providing access to the data repository, but is responsible for limiting access. For this reason, the information policy is tightly integrated with the data repository.

The information policy works with a session repository that stores session information associated with any collaboration similar to the structure shown in Figure 4.18. In addition, the information policy must exist so that any application can utilize the policy to enforce the protocols uniformly across all applications. The placement of the information policy, including a customized information policy component used for a particular application, is shown in Figure 4.19. The information policy exists at an abstract level to be defined by a particular implementation, allowing multiple information policies to be developed for a single system.

When a client application sends a transaction, the information policy determines if the client has the appropriate permissions to process the transaction, then passes the transaction on to the appropriate data repository or refuses the request and suggests alternatives to the client. When clients change their position in the session (speaking, whispering or requesting additional permissions), the information policy changes the state of the client on the system. A session repository is necessary to ensure the collaboration information is stored and logged. As new members enter the system, they will need access to information policy related information such as who is in the session, who is currently speaking and most importantly, who is in charge of the session. Utilizing a repository, the information policy component is able to make use of the collaboration manager component to notify clients when participants enter, leave or change status in the session.

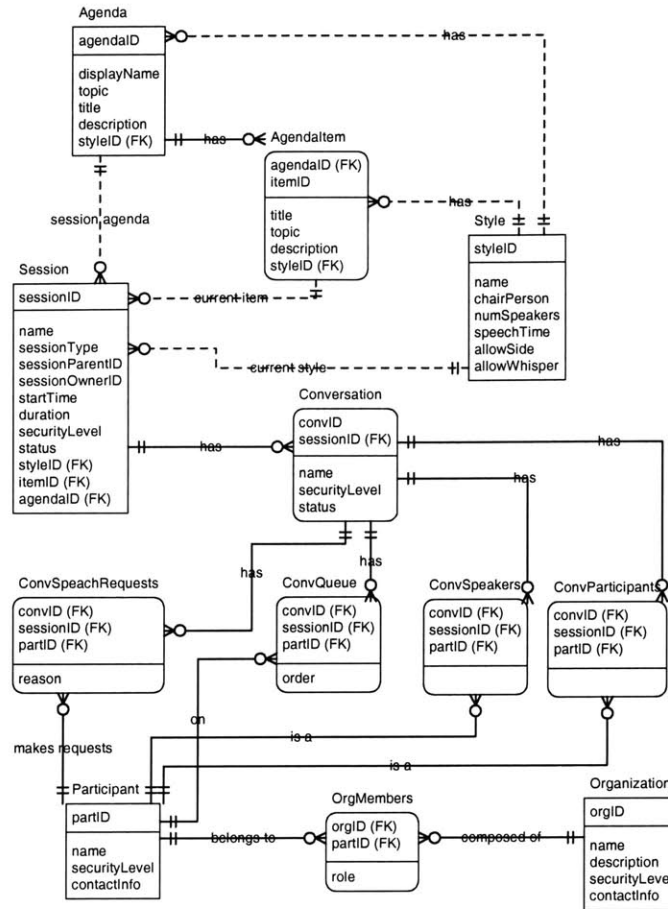


Figure 4.18: Database structure for collaboration session information

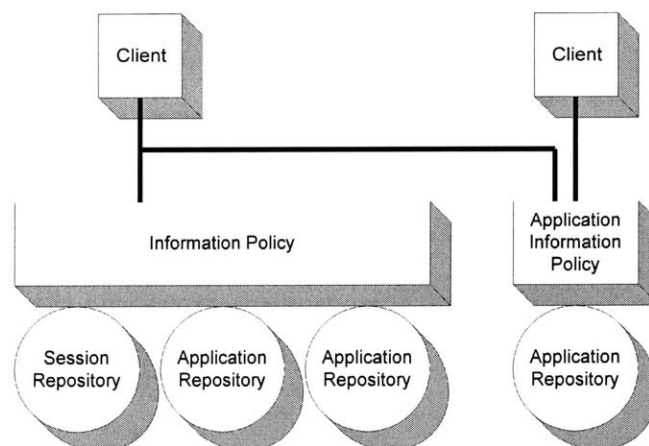


Figure 4.19: Information policy implementation

4.6.3 Data Repository

The data repository supports all aspects of the collaboration system. Although designed as a single component, the data repository is actually a set of loosely organized databases. Client applications access their repository through an interface, or repository manager, that directs requests to the appropriate data repository. The information policy component sends approved transactions to the manager to determine which repository to send the request to. Figure 4.20 shows the placement of the repository manager and the paths for transactions. Note that some transactions bypass the information policy and the repository manager, making direct calls to the repository. This flexibility is designed to allow client applications to bypass all layers for some transactions. For example, a system application may be used to mine data stored on a particular repository, however it has no need to use the collaborative features and can therefore bypass the system.

The implementation of the repository is assumed to be a database system. Database systems have higher performance and greater data persistency than other solutions, such as flat-files. Relational databases are web-enabled and include extensibility through embedded development languages like Java. Most object-oriented data types can be mapped to a relational database for persistency or can make use of an object-relational or object-oriented database. Separation of the data repository from the client access point allows databases to be swapped during a session and multiple databases to work together.

To accommodate latecomers, the data repository stores information for at least the duration of the collaboration session. When a participant joins a session late, it is rebuilt to the current state based on information in the repository. It is required to offer the option of persistency beyond the session for replay purposes, however some sessions choose to be “off the record”. These collaborations are not stored on the database. Reasoning behind this may be security or uncertainty of the validity of the data. An “off the record” session will operate similarly to the normal session. However, replay of the collaboration at a later time will not be possible. The data repository also provides the means for long-term storage of session information within the system.

The challenge of data storage can be addressed by using a commercially available

solution, or by creating a custom system. Many Internet database systems provide a scalable, robust solution to the collaboration system. Using an existing data storage solution leverages the power of third party products and expertise. Most commercially available database systems provide access via many development languages, such as ODBC, Java JDBC and C++. The implementation for the current system uses the Oracle 8i database, an object-relational database with embedded support for the Java programming language. This means the database can execute Java applications within itself, via standard SQL syntax. The prototype system calls a Java stored procedure when data is written to a table in the database, sending a message to the collaboration manager.

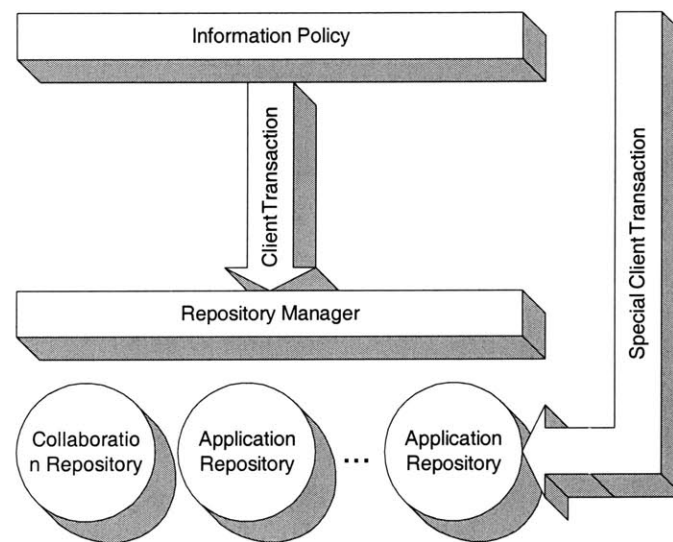


Figure 4.20 Repository manager and data repository

4.6.4 Collaboration Manager

Once a message arrives in the collaboration manager, it is sent to the appropriate client. The collaboration manager component is responsible for handling a single collaboration session, therefore, the whole system has several managers. Each session runs on its own manager to isolate itself and minimize security risks and fatal errors. If one session crashes on the system, others should not be affected. Within each manager component is the essential information to efficiently run a synchronous collaboration including participant and

session properties.

Within the collaboration manager, there is a collection of participants in the session who wish to receive notification when changes are made to the repository. When a new participant comes online and attempts to register with the collaboration manager, a check is done with the data repository to ensure the participant is allowed in a particular collaboration session. The collaboration manager also maintains a list of open conversations and their participants.

As a new message enters the collaboration manager from the repository, it is passed through various filters to determine the type of message and target conversation or participant. Determining who gets the message within the server minimizes network traffic and the load required by the client application. Some messages will need to be sent to all participants and all conversations, therefore, each collaboration session has a single conversation containing all participants. A message entering this conversation is sent to all participants registered with the collaboration manager. Some participants may not be actively monitoring the session conversation, requiring a session wide event to be sent to all participants regardless of the conversation they are actively monitoring to ensure successful message delivery. As an example, consider a side conversation involving two mission planners in a large collaboration session. While in the side conversation, an important decision is presented, requiring a response from everyone. The planners in the side conversation may not be monitoring the main conversation and may slow down the session unless they are notified. Sending the event message to all conversations allows the system to alert the side conversation to the important topic underway in the main conversation. They can then choose to ignore the event, end their side conversation, or just pause it for the time being.

The collaboration manager architecture is detailed in Figure 4.21. Notice that the list of participants contains more participants than are currently in the session. In the diagram, only Participants 3, 6 and 7 are utilizing the collaboration manager component of the system. This does not mean the others are not in the collaboration, it just means that they are not receiving events from the system when information in the repository is updated. Also note

that Participant 7 is in a whisper conversation alone, representing an independent investigation alone, perhaps refreshing himself/herself with the material being discussed. While in the whisper conversation, the participant can work with applications and not interrupt the rest of the session. When they wish to join the collaboration again, the client application can transfer information generated in the whisper conversation to the main conversation depending on the capabilities of the implementation of the client application.

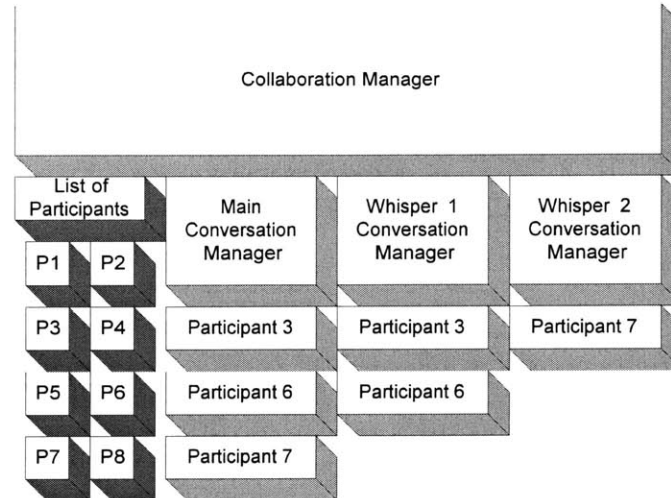


Figure 4.21: Collaboration manager architecture

4.6.7 Custom Components

To address the issues of scalability and future requirements, custom implementations of any component can be plugged into the system. Customizable components can perform specific functionality not in the existing system allowing them to work with other components at the interface level. In addition, custom collaboration managers can work in systems with default managers to handle the same collaboration session in different capacities. For example, consider a client application that requires secure communication on a wireless network using encryption through a special operations planning application using direct networking instead of the Internet (see Figure 4.22). This would require a custom collaboration manager capable of encrypting the message and sending it over the alternative network. Moreover, custom client applications may work with standard implementations of the data repository and information policy, requiring only the custom collaboration manager.

Another feature of the system shown in Figure 4.22 is the inclusion of the default collaboration manager, which serves clients on the Intranet or Internet while the custom manager serves those agents whose transmissions must be encrypted.

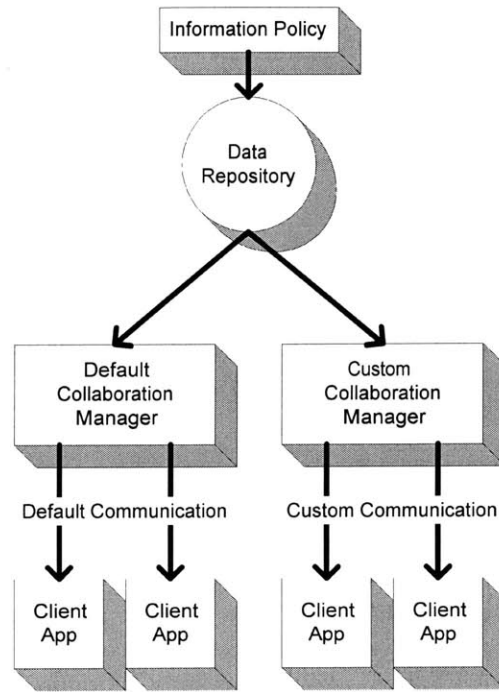


Figure 4.22: Custom implementation of system components

5. MULTIPLE DEVICE ACCESS

The two main components of the current trend of information technology enables people distributed geographically to communicate each other. One is that the data transfer rate of wired and wireless communication has increased to some extent. The other is that the downsizing client devices such as Handheld PCs, Personal Digital Assistants (PDAs) and Java-enabled phones can be used to facilitate especially field workers to communicate with other personnel and to access to data repositories.

This chapter features the current technologies that have potential for facilitating collaboration between teams that are distributed geographically and are working toward the common goal.

5.1 DESKTOP ENVIRONMENT

The core of any Internet enabled system is the desktop solution: a device that is fixed in position for a given session. Multi-device functionality exists within this realm, as there are many desktop devices that must communicate. The design of the Internet and related technologies has taken care of several issues, allowing Macintosh users to see the same Web content as a user on a UNIX machine. Portability from platform to platform is key in multi-user, multi-session collaborations, where participants are expected to join randomly with minimum hardware requirements. Desktop solutions need to be designed to allow devices of all configurations and capabilities to collaborate.

5.2 PORTABLE PC


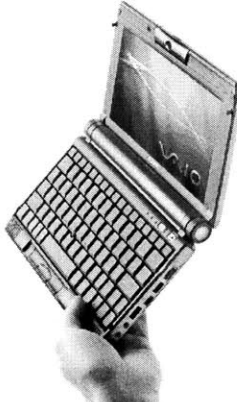
Portable computers are the transform of a typical desktop or fixed computer into a portable package. Table 5.1 shows specifications of currently available commercial products. CASIO MPC-206E Cassiopeia FIVA[19] and SONY VAIO PCG-C1VN[20] are outstanding products in terms of the size and battery life if compared with the other laptop PCs, and their performances are powerful enough and they have equivalent capabilities with ordinary desktop PCs. Their characteristics of small-sized, lightweight and long battery life

offers users a great opportunity to access PC resources in the field.

The limitations of these products stem from their small-sized bodies: the user interface that includes both screen size and keyboard size. It is impractical to use the same user interface as a desktop environment that usually requires XGA (1024x768 dots) or higher resolution. The differences of keyboards between a desktop PC and a small-sized portable PC are key pitch and keystroke, which discourages a user from typing a large document.

Portable PCs can be connected through wireless networks to a collaboration system allowing access to most of the functionality of desktop environments. These devices have capabilities similar to the desktop solution, requiring only a customized look and feel to work better with the hardware: a low resolution screen and a small keyboard.




Table 5.1: Portable PC

Model	MPC-206E Cassiopeia FIVA	VAIO PCG-C1VN
OS	Microsoft Windows ME	Microsoft Windows ME
CPU	Crusoe™ TM5600 (600MHz)	Crusoe™ TM5600 (600MHz)
RAM	128MB (max 192MB)	128MB (max 192MB)
HDD	20GB	12GB
Display	8.4" TFT color LCD SVGA (800x600 dots) 16bit color (65,536 colors)	8.9" TFT color LCD UWXGA (1024x480 dots)
Keyboard	80 keys 16 mm pitch/1.5mm stroke	86 keys 17 mm pitch/2mm stroke
Pointing Device	Stick type with 3 buttons	Stick-type compatible with scroll function and Jog Dial™
Interface	USB port PC card slot: Type II (Card Bus) CF card slot: Type II Modem port: RJ11 phone jack LAN port (100BASE-TX/10BASE-T) Stereo Headphone Microphone VGA output (via dedicated CASIO cable) Serial port: IEEE1394: 4-pin	USB port PC card slot: Type II (Card Bus) Modem port: RJ-11 phone jack headphone out audio in VGA output i.LINK® (IEEE 1394) S400 interface3 MagicGate™ Memory Stick®
Buttery life	4.5 hours (approx.) (with standard battery)	2.5 - 5.5 hours (with standard battery)
Weight	2.2lbs	2.2lbs
Dimensions (HxWxD)	0.83" x 8.78" x 7.77"(approx.)	1.06" x 9.8" x 6.0"
Image		

5.3 PERSONAL DIGITAL ASSISTANT

Personal Digital Assistants (PDAs) are more portable and less resources than portable PCs. The distinctive feature of a PDA that it is designed to be synchronized with PCs by default. A PDA has a capability of sharing with PCs and is equipped with applications similar to PC applications such as Microsoft Word and Microsoft Excel although the functionalities are limited. Also a PDA can access to LAN or WAN by using wired and/or wireless connections. Most of the commercial products are equipped with PC Card slot and/or Compact Flash (CF) Card slot that allow users to use networking devices such as LAN card, modem card and wireless phone connection card. Table 5.2 shows specifications of some of the commercial products.

Table 5.2: PDA

Model	iPaq H3650 Pocket PC	Cassiopeia E-115	Jornada 548
OS	Windows® Powered Pocket PC	Windows® Powered Pocket PC	Windows® Powered Pocket PC
CPU	206 MHz Intel StrongARM	131 MHz	133MHz 32-bit Hitachi SH-3
RAM	32MB	32MB	32MB
ROM	16MB	16MB	16MB
Display	TFT color LCD 240x320 dots 4096 colors	TFT color LCD 240x320 dots 65,536 colors	STN color LCD 240x320 dots 4096 colors
Input method	Handwriting recognition Soft keyboard Voice record Inking		Pen-and-touch interface Handwriting recognition On-screen keyboard 4 quick launch screen icons 2 quick keys
Interface	USB/Serial port Infrared port (115kbps) Speaker & Microphone Audio out jack	Serial port: RS232C Infrared port (115.2bps) CF card slot: Type I/II Speaker & Microphone Stereo earphone jack	RS232 serial port USB cradle IrDA infrared port CF card slot: Type I Speaker & Microphone Stereo earphone jack
Weight	6.3 oz.	9.0 oz.	9.1 oz.
Dimensions (HxWxD)	0.62"x 3.28"x 5.11"	3/4"x 3 1/4"x 5 1/8"	0.6"x 3.1"x 5.2"
Image			

5.4 JAVA-ENABLE PHONE

Java-enable phones allow users to interact with Web-based services as well as to execute functional stand-alone applications. Compared with PCs and PDAs, the resources are very limited: user interface including both input and output, processing power, memory, and storage. The application available to Java-enable phones must be designed carefully to meet with these constraints.

5.4.1 Java 2 Platform, Micro Edition

Java technology can provide mobile devices with advanced features. For example, cellular phones can interact with a back-end server dynamically. Sun has developed a standard for putting its Java software on to cellular phones, providing Java 2 Platform, Micro Edition (J2ME)[24] that consists of a virtual machine and a set of APIs suitable for providing tailored runtime environments for consumer and embedded electronics. Sun and Java Community have worked together to define two J2ME configurations. These configurations consist of core library sets and virtual machines optimized for the characteristics found in the two device groups: devices that are mobile, and devices that typically are fixed. The design center for the small handheld devices is addressed by J2ME Connected Limited Device Configuration (CLDC). CLDC specifies the core libraries and virtual machine features. The target devices are cellular phones, pagers, mainstream personal digital assistants, and small retail payment terminals.

The heart of J2ME technology in mobile devices (the CLDC) is Sun's K virtual machine (KVM). The KVM is a virtual machine designed from the ground up with the constraints of inexpensive mobile devices in mind. Named to reflect that its size is measured in the tens of kilobytes, the KVM is suitable for devices with 16/32-bit RISC/CISC microprocessors/controllers, and with as little as 160 K of total memory available – 128 K of which is for the storage of the actual virtual machine and libraries themselves.

The Mobile Information Device Profile (MIDP) is a set of Java APIs, which together with CLDC, provides a complete J2ME application runtime environment. The MIDP

specification addresses issues such as user interface, persistence storage, networking, and application model.

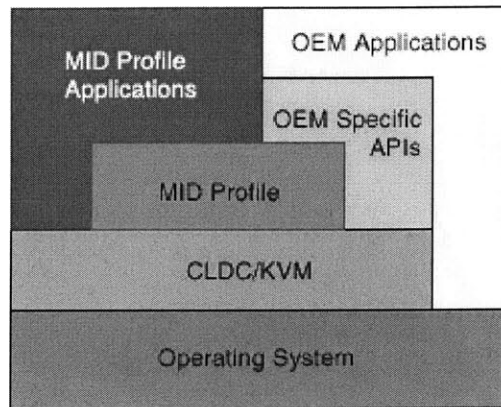


Figure 5.1: J2ME wireless architecture [24]

5.4.2 Java-enabled Phone Service




NTT DoCoMo[25], the dominant wireless carrier in Japan, released Java-enabled phones on January 26, 2001. The platform of DoCoMo's Java-enabled phone is based on CLDC specification and contains additional libraries developed by DoCoMo and device manufacturers. The libraries developed by a device manufacturer allow applications to make use of device-specific resources. The Java application designed for DoCoMo service, called "i-appli", is restricted in its size and its working memory: the "i-appli" should be less than 10KB in the form of .JAR file; its working memory is less than 5KB.

J-Phone[26], one of the major wireless carriers in Japan, will start the Java contents services in June, 2001. J-Phone adopted CLDC and MIDP as the platform and added J-PHONE Special Device Libraries (JSCL). The Java application for J-Phone service is allowed to be up to 30KB and the Java platform supports 3D objects.

Due to the size of the body, user interface is very limited as well as processing power, memory, and data storage. Some of specifications of the Java-enabled phones released by DoCoMo are described in the Table 3.3. QVGA (quarter VGA) display restricts the amount of information to be displayed. Also user input is restricted. An ordinary cell phone contains

about twenty keys including 4-direction navigation key that are closely arranged in a small area.

Table 5.3: Java-enabled phone

Model	SO503i	N503i	P503i
Manufacturer	SONY	NEC	Panasonic
Display	TFT color LCD 120x160 dots 65,536 colors 31.5x40.5mm	TFD color LCD 120x160 dots 4096 colors 30.5x40.5mm	STN color LCD 120x160 dots 256 colors 28.5x38mm
Weight	115g	98g	74g
Dimensions (HxWxD)	98x49x28mm (when folded)	93x48x22mm (when folded)	128x45x17mm
Image			

5.4 WIRELESS COMMUNICATION

Wireless networks can be categorized into three types according to its geographical coverage:

- WAN: Wide Area Network, which is normally covers a city, a nation, even continental region;
- LAN: Local Area Network, which serves in a room, in a building, typically within a organization;

- PAN: Personal Area Network, which is effective only in very close area like inside a building.

5.4.1 Wireless WAN

The wireless data industry is still working out the kinks in developing and building data transmission networks and agreeing on standards and protocols. There is more than one competing visions. Experts believe that the growing market for wireless data will support multiple networks and protocols – and hi-speed broadband networks.

- Global System for Mobile (GSM) communications, which is based on TDMA (Time Division Multiple Access). GSM is a digital cellular or PCS standard used throughout the world and de facto standard in Europe.
- Cellular Digital Packet Data (CDPD) is a packet data protocol designed to work over AMPS (the original cellular network) or as a protocol for TDMA.
- Code Division Multiple Access (CDMA), a spread spectrum air interface technology used in some digital cellular, personal communications services and other wireless network.

Wireless-capable Internet access can be classified into three broad categories:

- Full user mobility – Examples include cellular digital packet data (CDPD), in which users can access data while on the move, such as in a car. Yet, Internet access via fully mobile systems occurs at relatively slow data throughput speeds.
- Portable wireless data – Such networks let users access the Internet while they are in the coverage area, using a laptop or handheld PC, and a small wireless modem. But these networks do not offer full mobility during online sessions. In other words, if a user tries to access the Internet while in a moving car, the connection may drop. Speeds are generally higher than with fully mobile systems but are still relatively slow when compared to the current dial-up modem technology.

- **Fixed wireless data** – This offers service to a location, such as an office or home, through larger customer-premises antennas than seen in the mobile or portable setups. The fastest data throughputs – up to T-1 speed – are available over fixed wireless networks. Examples include Teligent and wireless T-1.

Wireless networks face great challenges as well. The surrounding environment interacts with the signal, blocking signal paths and introducing noise and echoes. As a result, wireless communication is characterized by lower bandwidths, higher error rates, and more frequent disconnections. These factors can in turn increase communication latency resulting from retransmissions, retransmission time-out delays, error-control protocol processing, and short disconnections. Mobility can also cause wireless connections to be lost or degraded. Users may travel beyond the coverage of network transceivers or enter areas of high interference. Unlike typical wired networks, the number of devices in a network cell varies dynamically, and large concentrations of mobile users at conventions and public events, may overload network capacity.

5.4.2 Wireless LAN

Worthy of noticing also is the wireless LAN technology, in which a wireless LAN access point is connected to the wired LAN and responsible for communicating with desktop PCs, laptops or devices that have wireless connectivity via radio frequency (RF) technology. Wireless LAN is useful in hotspot areas, like airports, health-care centers, warehouses, etc.

5.4.3 Wireless PAN

PAN is short-range wireless network for personal communications. The underlying technology is called Bluetooth, a radio-based technology that enables devices to communicate at close range, without the need for direct line of site or additional communication protocols. It has gained much attention in the industry because of its potential ability to let cell phones communicate with PCs and PDAs. Bluetooth has been in the development stage for more than a year and still faces technological hurdles.

6. ROBOT-HUMAN INTERACTION

6.1 OVERVIEW

To locate victims is one of the key search and rescue (SAR) activities, but SAR teams often find it difficult to locate victims for some reasons:

- A collapsed facility/building has no space where SAR personnel can enter.
- A damaged building is about to fall down.
- A facility/building contains a potential danger of a gas leaks or hazardous materials spills.

In these situations, SAR dogs and their handlers play an integral role in locating victims. Several limitations and constraints exist in SAR canine teams. The feedback from dogs contains very limited: dog-handler interaction relies on barking alert skills. The canine/handler team must pass rigorous certification that usually takes years. The biggest problem of SAR dogs is a quarantine: in a large-scale international disaster relief mission, SAR dogs from abroad may not allowed to enter the disaster area.

A robot can be an alternative solution for SAR dogs. This chapter discusses about human-robot interaction and focuses on requirements of SAR robots. Also some of the current technology issues are described.

6.2 REQUIREMENTS

The main goal of a SAR robot is to collect information on a hostile environment where a field agent cannot enter and to send feedback to its operator. The functionalities of a robot are categorized three parts: ability to move around a hostile environment; ability to sense surroundings and store information; and ability to communicate with its operator.

In a disaster area, a robot will be exposed to a harsh environment and move around a wobbly surface so that a robot needs to be designed under some limitations. For example, power supply will be limited so that a long battery life is required. One of the important features of a SAR robot is that robot can move around both under the proper command control of its operator and under its autonomous control. Due to the nature of a disaster area, a robot will often be out of range of its operator's control. In that situation, a robot must have an ability to move around by itself, sense surroundings, store information, and return to its operator.

A SAR robot should sense surroundings and store information so that a field agent or other relief mission personnel can know the site-specific information. Still images, motion picture and sound files will be useful for human to understand the situation intuitively. Also temperature or other sensors will be used for more detail analysis. Information that keeps track of a route of a robot is important to locate the events a robot comes across.

The communication between a robot and its operator consists of the operator's command control and a robot's feedback. A robot feedback contains its status and sensing information. An operator denotes the same personnel as a field agent in this study, and will have a handheld device such as a PDA and a Java-enabled phone. To make a seamless connection between an operator and a robot, a robot can be connected to an operator's client device wirelessly using either infrared or radio transmitters.

6.3 LEGO MINDSTORMS

MINDSTORMS™ is the name of a product line sold by the LEGO® Group. The centerpiece of MINDSTORMS is the Robotics Invention System (RIS), which is a set for building robots. A set consists of a robot brain called the RCX which is a small computer based on a Hitachi H8 series microprocessor, several kinds of sensors, two motors and LEGO pieces that a user can use to build the body of a robot. The set contains an IR (infrared) tower that attaches to one of the serial ports on a PC so that a user can download programs.

6.3.1 Programming Environment

The RCX is a small computer which contains an 8-bit CPU, serial I/O, Analog to Digital conversion, built-in timers, 16K of internal ROM which is programmed with some lower-level operating system software, and 32K of static RAM. 6k of RAM is reserved for user memory in which the program a user write for the RCX reside is stored. The RCX operating system is split into two parts: the first part is stored in ROM; the second part is stored in PAM and must be downloaded to the RCX the first time it is turned on.

MINDSORMS offers the RIS programming environment, RCX Code that has a graphical user interface and is designed for a user who has no or little programming experience so that a user can write programs easily but to access to the resource is limited. Several alternative programming environments are developed by third-party groups.

6.3.2 Input: Sensors

MINDSTORMS Robotics Invention System includes a various types of sensors:

- Touch sensor
- Light sensor
- Rotation sensor (optional)
- Temperature sensor (optional)

6.3.3 Output

MIDSTORMS RCX contains three output ports: usually they are used to manipulate its movement by controlling motors.

6.4 JINI TECHNOLOGY

Jini™ technology provides simple mechanisms that enable devices to plug together to form an impromptu community that is put together without any planning, installation, or human intervention. Each device provides services that other devices in the community

may use. These devices provide their own interfaces, which ensures reliability and compatibility.

The Java™ programming language is the key to making Jini technology work. Devices in a network employing Jini technology is tied together using Java Remote Method Invocation (RMI). The discovery and join protocols, as well as the lookup service, depend on the ability to move Java objects, including their code, between Java virtual machines (JVMs).

Jini technology not only defines a set of protocols for discovery, join, and lookup, but also a leasing and transaction mechanism to provide resilience in a dynamic networked environment. The underlying technology and services architecture is powerful enough to build a fully distributed system on a network of workstations. And the Jini connection infrastructure is small enough that a community of devices enabled by Jini connection software can be built out of the simplest devices.

7. CONCLUSION

In a large-scale disaster relief mission, a variety of personnel and organizations with different expertise take part in the relief activities towards a common goal. Interactions between them are important to promote efficiency, and the relief activities, especially search and rescue (SAR) activities, are carried out in a short time period.

Some Emergency Offices of City/State/Federal governments and two case studies of actual disaster relief activities after large earthquakes were examined in order to recognize the issues in terms of collaborative environments during a disaster relief missions. These examinations have revealed the fact that the availabilities of site-specific information is a major concern when some decisions concerning relief activities are made, but the communication between on-site personnel and headquarters is not enough mainly due to the lack of cross-organizational coordination.

Several commercial solutions have been developed and several studies have been made on collaborative environments at MIT. The current typical solutions consist of e-mail system, group discussion, bulletin board system, and shared data repositories integrated with mapping systems. Some further enhancements of these systems should be considered: multiple client access and detail design of thin portable client devices.

To develop collaborative environments for disaster relief missions, the requirements have been gathered and the requirement analysis has been developed. According to functional role, relief mission participants are classified into four groups: field agent, team leader, coordinator, decision maker and specialist/advisor. The category “field agent” plays an integral role during a mission so that its activities are focused on. The hierarchy of collaborative session has been developed on account of that of participant roles.

The multiple device access is an important functionality of collaborative environments. Some of the leading portable devices are featured as well as desktop environments, and three types of wireless communication are described. Among various portable devices, a Java-enable phone deserves careful attention since it is the most portable Internet-appliance

although it has very limited resources.

To collect and report site-specific information in a disaster area is critical to make decisions how much resources including both staff and equipment to deploy. The needs and requirements of robot-human interaction that can be applied to a relief mission are discussed in comparison with SAR dogs. Some of the state-of-the-art technologies such as robots and network connections between devices are reviewed.

BIBLIOGRAPHY

- [1] Federal Emergency Management Agency, <http://www.fema.gov/>
- [2] City and Country of San Francisco, <http://www.ci.sf.ca.us/>
- [3] New York City, <http://www.nyc.gov/>
- [4] Hyogo Prefecture Policy Planning and Administration Dept. Disaster Management Bureau (1995), *Record of the first month after the Great Hanshin-Awaji Earthquake*, <http://web.pref.hyogo.jp/syoubou/daishinsai/ikkagetu.html> (in Japanese)
- [5] Kobe City Fire Bureau, <http://www.city.kobe.jp/cityoffice/48/> (in Japanese)
- [6] KOBEnet, <http://incede.iis.u-tokyo.ac.jp/kobenet/>
- [7] The United States Agency for International Development (1999), Turkey Earthquake Fact Sheet #29, http://www.usaid.gov/hum_response/ofda/turkeq_fs29.html
- [8] Earthquake Engineering Research Institute, The Izmit (Kocaeli), Turkey Earthquake of August 17, 1999, *EERI SPECIAL EARTHQUAKE REPORT – October 1999*, <http://www.eeri.org/Reconn/Turkey0899/Turkey0899.html>
- [9] Bogazici University Kandilli Observatory & Earthquake Research Institute, <http://www.koeri.boun.edu.tr/earthqk/earthqk.html>
- [10] Essential Technologies, Inc., <http://www.essentech.com>
- [11] California's Response Information Management, <http://www.rimsinland.oes.ca.gov/>
- [12] CSMCo, <http://www.csmco.com/>
- [13] Eteam.com, Inc., <http://www.eteam.com/>
- [14] Lotus, <http://www.lotus.com/>
- [15] The DaVinci Initiative, Massachusetts Institute of Technology, <http://ganesh.mit.edu>
- [16] Hussen, K. (1998), Communication Facilitators for a Distributed Collaborative Engineering Environment, Ph.D. Thesis MIT.
- [17] The Charles Stark Draper Laboratory, Inc., <http://www.draper.com/>
- [18] Peña-Mora, F., Hussein, K., Vadhavkar, S., and Benjamin, K. (2000), CAIRO: a concurrent engineering meeting environment for virtual design teams, *Artificial Intelligence in Engineering, Volume 14, Issue 3*, pp.203-219.
- [19] CASIO, MPC-260E Cassiopeia FIVA, <http://www.casio.com/personalpcs/product.cfm?section=17&product=3917&display>

=10&cid=4841, April 2001.

[20] SONY, VAIO PCG-C1VN, <http://www.sonystyle.com/vaio/picturebook/specs.htm>, April 2001.

[21] Compaq, iPaq H3650 Pocket PC, <http://www.compaq.com/products/handhelds/pocketpc/h3650b.shtml>, April 2001.

[22] CASIO, Cassiopeia E-115, <http://www.casio.com/personalpcs/product.cfm?section=19&product=1714&display=10&cid=3222>, April 2001.

[23] HP, Jornada 548, <http://www.hp.com/jornada/products/540/overview.html>, April 2001.

[24] Sun Microsystems, <http://java.sun.com/j2me/>

[25] NTT DoCoMo, <http://www.nttdocomo.com/>

[26] J-PHONE, <http://www.j-phone.com/>

[27] Knudsen, J. B. (1999), *The Unofficial Guide to LEGO® MINDSTORMS™ Robots*, O'Reilly & Associates, Inc., Sebastopol, CA.

[28] Baum, D. (2000), *Dave Baum's Definitive Guide to LEGO® MINDSTORMS™*, APress, Emeryville, CA.

[29] Baum, D., Gasperi, M., Hempel, R., and Villa, L. (2000), *Extreme MINDSTORMS™: an Advanced Guide to LEGO® MINDSTORMS™*, APress, Emeryville, CA.